

# A Pragmatic View of MANET Performance Evaluation

and

Design of a Prototype MAC Level Routing Algorithm

by

Michael J. Thurston

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE


in partial fulfillment of the requirements for the

Degree of Master of Science


in

Electrical and Computer Engineering

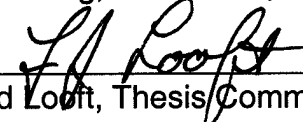
by

  
January 2003

APPROVED:

  
\_\_\_\_\_  
Dr. David Cyganski, Thesis Advisor

  
\_\_\_\_\_  
Dr. Brian King, Thesis Committee

  
\_\_\_\_\_  
Dr. Fred Looff, Thesis Committee

## **Abstract**

Our goal in this research is to investigate and determine how to best support a challenging mobile wireless network based in a military operational environment. Since routing protocols used in mobile ad hoc networks (MANET) must adapt to frequent or continual changes of topology, while simultaneously limiting the impact of tracking these changes on wireless resources, we focused our initial research on improving the efficiency of route discovery. We proposed and designed a new MAC layer routing protocol that pursues reduced routing overhead, greater interaction of network protocol layers and passive neighbor/path discovery. This algorithm, called Virtual MAC Tag Switching (VMTS), evolved as we implemented a prototype in the ns-2 network simulator and conducted simulation analysis of existing protocols: DSDV, DSR and AODV. Upon analyzing the performance of existing routing protocols using pragmatic metrics not applied in any MANET literature it was found that current MANET models produce unsatisfactory performance. Subsequent analysis of transport layer protocol behaviors pinpointed the causes that undermine the performance of the existing protocols and would have thwarted VMTS as well.

## **Acknowledgements**

I would first like to acknowledge the United States Army and the Army Acquisition Corps for sponsoring me in pursuit of the Degree of Masters of Science in Electrical Computer Engineering and subsequently this research. Of significant mention is the guidance and motivation provided to me by my thesis advisor Professor David Cyganski. His professionalism, expertise, determination and compassion have been an inspiration to me personally and professionally. His unique methodology has challenged not only my academic aptitude but also my leadership ability.

I would also like to express my deepest gratitude to the VMTS implementation team for taking on the task of coding VMTS and modifying ns as a course project despite the foreknowledge that the demands of this endeavor would far exceed that of the standard High Performance Networks course project. Aastha Kathuria, Abhishek Anand Kumar, Ritesh Shukla, Roshan John, Shiwangi Deshpande, Shobha Sharoff Setty, Vineet Sharad Agrawal and Yogesh Chandrakant Samant, you decided to take the road less traveled and in turn have made my academic experience more rewarding as I hope and believe you have made your own as well. Additionally, I want to thank David Holl for sharing his expertise in the wonderful world of ns.

Finally and most importantly, I would like to honor my wife Anne and my five children for their understanding, patience and support while I pursued yet another chapter of my personal and professional development.

## Table of Contents

|  |    |
|--|----|
| Abstract.....  | i  |
| Acknowledgements .....   | ii |
| List of Tables.....  | v  |
| List of Figures.....   | v  |
| 1 Introduction .....   | 1  |
| 2 Mobile ad hoc networks .....                                 | 3  |
| 2.1 Applications of MANETs .....                               | 4  |
| 2.2 MANET Internet Engineering Task Force .....                | 4  |
| 2.3 Characteristics of MANETS .....                            | 5  |
| 2.4 Functions of MANET routing protocols .....                 | 7  |
| 2.4.1 Topology/route discovery and maintenance .....           | 7  |
| 2.4.2 Economy of resources .....                               | 8  |
| 2.4.3 Interaction with adjacent layers.....                    | 8  |
| 3 Operational Environment and Factors .....                    | 9  |
| 3.1 Organizational type and structure .....                    | 9  |
| 3.2 Unit Size and Corresponding Node Quantities .....          | 9  |
| 3.3 Battlespace Dimensions and Expected Nodal Dispersion ..... | 10 |
| 3.4 Target Unit Level .....                                    | 12 |
| 3.5 IP Address Assignment Strategies .....                     | 12 |
| 4 Wired Network Techniques.....                                | 14 |
| 4.1 Local Area Networking (LAN) .....                          | 14 |
| 4.2 Internetworking .....                                      | 14 |
| 4.3 Bridging.....  | 15 |
| 4.4 LAN Switching .....  | 15 |
| 4.5 Routing .....  | 16 |
| 4.6 IP over Ethernet.....                                      | 16 |
| 4.7 Wireless adaptation .....                                  | 17 |
| 5 VMTS protocol description.....                               | 19 |
| 5.1 General characteristics .....                              | 20 |
| 5.2 VMTS routing.....  | 21 |
| 5.2.1 Route discovery .....                                    | 21 |
| 5.2.2 Message forwarding .....                                 | 22 |
| 5.2.3 Route repair .....                                       | 22 |
| 5.2.4 Neighbor re-associate.....                               | 23 |
| 5.2.5 Neighbor discovery .....                                 | 23 |
| 5.2.6 ARP requests and broadcast messages.....                 | 23 |
| 5.2.7 Full path discovery (FPD) .....                          | 24 |
| 5.3 Promiscuous mode .....                                     | 24 |
| 5.4 Routing table.....   | 25 |
| 5.4.1 Table fields .....                                       | 25 |
| 5.4.2 Null fields .....  | 25 |
| 5.4.3 Table lookup procedure .....                             | 26 |
| 5.4.4 Table write procedures .....                             | 26 |

|       |  |    |
|-------|--|----|
| 5.4.5 | Multiple entries per destination .....                 | 29 |
| 5.5   | VMTS MAC Frame.....                                    | 29 |
| 5.5.1 | MAC_TAG .....  | 30 |
| 5.6   | VMTS protocol routing example.....                     | 31 |
| 6     | Simulation environment.....                            | 36 |
| 6.1   | ns-2 simulator .....                                   | 36 |
| 6.1.1 | Physical layer.....                                    | 36 |
| 6.1.2 | Network interface.....                                 | 37 |
| 6.1.3 | MAC layer .....  | 37 |
| 6.1.4 | Link layer .....                                       | 37 |
| 6.1.5 | Mobile node object.....                                | 38 |
| 6.2   | VMTS Implementation decisions .....                    | 40 |
| 6.2.1 | 802.11 MAC distributed coordination function .....     | 40 |
| 6.2.2 | ARP and source broadcast messages.....                 | 41 |
| 6.2.3 | Adjustable parameters.....                             | 41 |
| 6.3   | Description of ad hoc protocols implemented in ns..... | 43 |
| 6.3.1 | DSDV.....  | 43 |
| 6.3.2 | DSR .....  | 44 |
| 6.3.3 | AODV .....   | 44 |
| 7     | Simulation Methodology.....                            | 46 |
| 7.1   | Simulation topography .....                            | 46 |
| 7.2   | Movement model .....                                   | 46 |
| 7.3   | Communication/traffic model .....                      | 47 |
| 7.4   | Metrics .....  | 49 |
| 7.5   | Coding .....   | 50 |
| 7.6   | Verification and Validation .....                      | 51 |
| 7.6.1 | Verification.....                                      | 51 |
| 7.6.2 | Validation.....  | 51 |
| 8     | Routing protocol simulation results .....              | 55 |
| 8.1   | Network throughput.....                                | 55 |
| 8.2   | Standard deviation of source throughput .....          | 59 |
| 8.3   | Routing Overhead.....                                  | 60 |
| 8.3.1 | Overhead in packets.....                               | 60 |
| 8.3.2 | Overhead in bytes.....                                 | 63 |
| 8.4   | Packet delivery ratio.....                             | 63 |
| 9     | TCP performance and analysis.....                      | 65 |
| 9.1   | Network Throughput observations .....                  | 65 |
| 9.2   | Variability of source throughput .....                 | 68 |
| 9.3   | Interaction of protocol layers.....                    | 71 |
| 10    | Conclusions.....                                       | 73 |
|       | REFERENCES.....  | 75 |

## List of Tables

|  |    |
|--|----|
| Table 3-1 Node Quantities and Geographic Area by Unit Type .....               | 10 |
| Table 7-1 Effect of TCP packet size on throughput (2 node).....                | 48 |
| Table 7-2 Effect on TCP Packet size (50 node) .....                            | 49 |
| Table 8-1 Effects of transmission range on performance (50-node, 50-src) ..... | 57 |
| Table 9-1 Throughput of CBR sources varying offered load .....                 | 69 |

## List of Figures

|   |    |
|---|----|
| Figure 3-1 Squad Level Node Dispersion .....                                      | 11 |
| Figure 5-1 Node Link Connectivity Diagram.....                                    | 19 |
| Figure 5-2 Node Bridge Topology Diagram.....                                      | 20 |
| Figure 5-3 VMTS Routing agent process .....                                       | 21 |
| Figure 5-4 Node A routing table .....   | 25 |
| Figure 5-5 Table lookup procedure .....   | 26 |
| Figure 5-6 Partial table write procedure .....                                    | 27 |
| Figure 5-7 Full table write procedure.....  | 28 |
| Figure 5-8 Route discovery table write procedure.....                             | 29 |
| Figure 5-9 Generic 802.11 MAC frame .....   | 29 |
| Figure 5-10 VMTS MAC frame.....   | 30 |
| Figure 5-11 VMTS MAC_Tag.....   | 30 |
| Figure 6-1 Radio propagation model.....   | 37 |
| Figure 6-2 ns MobileNode objects (reprinted from <i>The ns Manual</i> [FAL])..... | 38 |
| Figure 6-3 VMTS MobileNode object .....   | 39 |
| Figure 7-1 DSDV packet delivery ratios as published in [BRO].....                 | 52 |
| Figure 7-2 DSDV PDR in 50-node validation trials.....                             | 53 |
| Figure 7-3 Protocol PDR comparison in 50-node, 10 source validation trials.....   | 53 |
| Figure 7-4 Protocol PDR values as published in [BRO] .....                        | 54 |
| Figure 8-1 DSDV avg. network throughput (50-node, 1Mbps links) .....              | 56 |
| Figure 8-2 AODV avg. network throughput (50-node, 1Mbps links).....               | 57 |
| Figure 8-3 DSR avg. network throughput (50-node, 1Mbps links).....                | 58 |
| Figure 8-4 Comparison of 50-source throughput (50-node, 1Mbps links) .....        | 58 |
| Figure 8-5 Comparison of source throughput std. dev. (50-node, 1Mbps links)..     | 60 |
| Figure 8-6 DSDV routing overhead (packets) (50-node, 1Mbps links).....            | 61 |
| Figure 8-7 AODV routing overhead (packets) (50-node, 1Mbps links).....            | 62 |
| Figure 8-8 Comparison of routing packet overhead (50-node, 1Mbps links).....      | 62 |
| Figure 8-9 Comparison of routing byte overhead (50-node, 1Mbps links).....        | 63 |
| Figure 8-10 Comparison of PDR with 50 TCP sources (50-node, 1Mbps links)..        | 64 |
| Figure 9-1 Interference tests .....   | 66 |
| Figure 9-2 Effects of broadcast message on channel .....                          | 67 |
| Figure 9-3 CBR and TCP PDR per offered load and source density.....               | 70 |
| Figure 9-4 TCP and CBR fairness per offered load and source density.....          | 71 |

# 1 Introduction

The purpose of this research is to investigate and determine how to best support a challenging mobile ad hoc network (MANET) wireless environment through improved routing strategies and quality of service enhancements; and to accomplish this with minimized routing overhead and efficient use of network resources.

This thesis will introduce the benefits of ad hoc networking, describe the characteristics of MANETs and identify the challenges faced when implementing routing schemes in support of MANETs. MANET protocols must perform the same functions of their wired counterparts but also perform functions specifically related to the MANET challenges. We describe these functions so that we have a better understanding of how our routing strategy would meet the demands of a MANET.

The environment in which a MANET is placed has a significant impact on the success of the routing strategy. Therefore, we chose to base our concepts and analysis on the assumption that we must support what is arguably the most demanding MANET environment, a tactical military environment.

In the process of developing our strategy, we examined those techniques that work in a wired network and determined how they can be implemented in a wireless network. We describe these networking techniques in general and point out those that could become the basis for a new protocol design.

We focus our initial research on MANET routing strategies but only after researching an expansive set of MANET related fields of study. We initially propose and design a MAC layer routing protocol, called Virtual MAC Tag Switching (VMTS), which is inspired by wired network bridging techniques. In designing VMTS, we pursued an algorithm with reduced routing overhead, greater interaction of network protocol layers and passive neighbor/path discovery. In conjunction with the protocol development, we analyze the performance of existing routing protocols, DSDV, DSR and AODV, using the ns-2 network simulator. We performed practical assessments incorporating a communication model not found in any papers and one that revealed how current ad hoc network models perform unsatisfactorily in true tests of network performance. During assessment of the ns simulations, we discovered startling results regarding these performance measures irrespective of the routing protocol used. All the while, the VMTS algorithm continued to evolve, particularly as issues arose during the prototype's implementation in ns. However, the results from the simulation of other protocols were so significant that it prompted us to stop further implementation of VMTS for even in its evolved state, VMTS would likely suffer much the same performance maladies (which ultimately was the case).

We turned our efforts toward subsequent analysis of transport layer protocols, fairness and congestion to analyze this unsatisfactory performance in an effort to understand the underlying causes. We find, as a result, that it is not the accuracy and efficiency of routing protocols that affect the true measures of

performance such as throughput or utilization, but the general notion of least distance routing itself.

This paper is organized as follows: Section two introduces mobile ad hoc networks and MANET applications. It describes MANET characteristics and identifies important functions that MANET protocols should perform. In section three, we provide details on the military operational environment and factors that influenced our design decisions. Section four describes the proven wired network techniques that we adapted to the wireless environment. Section five introduces our routing strategy and describes in detail the protocol we developed. Our simulation environment is illustrated and the methodology detailed in sections six and seven. We present results of our simulation of existing protocols in section eight and provide additional analysis and observations in section nine. The paper is concluded in section ten.



## 2 Mobile ad hoc networks

The information exchange industry, which includes wired, wireless and data exchange, has annual revenue on the order of a few trillion dollars. AT&T, the primary wired telephone service provider in the United States in the early 1980's had an annual budget equivalent to the budget of the fifth largest country in the world. Today, wireless industry income has surpassed that of wired telephone service and has achieved this in little more than a decade. Fueling this frenzy is the huge demand for wireless voice services found in today's cellular networks, which supports nearly a billion subscribers. At the same time, data services have also exploded with the rapid expansion of the Internet [PAH]. As both of these services infiltrate our every day lives it is understandable that consumers, commerce and business would desire wireless data services. Many attempts have been made to provide wireless wide area data networks, but most have not survived. IMT-2000, the organization driving next generation wireless standard development, has taken on the task to integrate, at least partially, data and voice wide area wireless networks. These networks are characterized by large infrastructure, which include thousands of base stations and many miles of cable to interconnect them. Mobile users access the network by operating in the coverage area or "footprint" of the base station which in turn routes the data packets or voice stream through a wired backbone to the appropriate destination.

A less pronounced area of wireless networks that is recently gaining attention is that of Mobile Ad Hoc Networks (MANET). MANETs are typically small in scale and do not require the infrastructure that the next generation cellular networks need to operate. In fact, there are no base stations in an ad hoc network. Therefore, they are less costly in time and money to install and operate.

Cellular and fixed mobile data networks can be called single hop networks since packets from the mobile terminal make only one wireless "hop" to its destination, the base station. Since MANETs do not have base stations a packet may take several wireless hops to reach its destination, therefore MANETs are known as multi-hop networks.

MANETs are dynamic, loosely organized networks whose members, called nodes, arbitrarily enter, exit and move around the network architecture [BRO][COR2][SU]. This arbitrary nodal motion can cause rapid, near random changes in the network topology.

MANET nodes are logically a router and may contain one or several hosts and communication devices [BRO][COR2]. Nodes forward packets for other mobile nodes that are not within the transmission range of the packet's source [BRO]. Nodes may consist of physically separate network devices [COR2] such as processors, hubs or bridges and a networking receiver-transmitter or they may be integrated into a single device such as a palm sized personal information system.

## 2.1 Applications of MANETs

New wireless applications are increasing as the technology, protocols and imaginations of entrepreneurs advance. Wireless technologies in areas such as intelligent antennas [ZYS], advanced coding techniques, signal processing, power management, battery size and lifetime, computer processing and modeling continue to mature. Couple this with research into bandwidth and power efficient protocols and investment into market research, wireless applications are moving beyond cell phones and local area network bridging.

MANETs are faced with considerable challenges due to their infrastructure-less architecture, mobility and multi-hop routing. Current MANET technologies provide relatively low capacities so they cannot yet provide the high-speed wide area services of their infrastructure based counterpart. This however doesn't mean that MANET technology won't occur at network edges where wired networks are too costly [COR2]. There will also be specialized applications where MANETs are the only possible solution.

MANET applications for distributed computing [SU] can be found in conference rooms and facilities, on campus, in corporate office space, and in industrial workspace. Emergency services applications are also evident, particularly in large-scale emergencies such as at the World Trade Center or in disaster relief situations [BRO][SU]. These scenarios are characterized by a massive mobilization of personnel from emergency workers, security, utility services, construction and demolition into an area with little or no power and a devastated infrastructure.

Military applications for MANETs are probably the most pervasive. Images of squads of soldiers equipped with personal area networks linked together by low power radios create dynamic MANETs as they patrol city streets or mountainous terrain. Integration with infrastructure based wireless systems form hybrid networks to get front-line images to the decision makers or fire support units. Larger scale applications involve high-speed forces maneuvering in battle sharing intelligence, operational and logistics information, to give each fighter and commander a common operational picture. The United States is incorporating applications such as these into today's forces as part of the Army's Transformation and is including broader capabilities in its concepts for the Objective Force.

## 2.2 MANET Internet Engineering Task Force

In 1997, the Internet Engineering Task Force (IETF) established the MANET Working Group (WG) within the Routing Area of the IETF. The MANET WG is largely a research organization due to the lack of commercial involvement. [BAK] The following excerpt from their IETF homepage describes their view of a MANET and the purpose of the WG:

"A "mobile ad hoc network" (MANET) is an autonomous system of mobile routers (and associated hosts) connected by wireless links--the union of which

form an arbitrary graph. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet.

The primary focus of the working group is to develop and evolve MANET routing specification(s) and introduce them to the Internet Standards track. The goal is to support networks scaling up to hundreds of routers. If this proves successful, future work may include development of other protocols to support additional routing functionality. The working group will also serve as a meeting place and forum for those developing and experimenting with MANET approaches." [IETF]

This description implies that MANETs are going to be independent or "closed" networks possibly with an interface with a larger Internet. It is also stated in MANET RFC 2501 Performance Issues and Evaluation Criteria that the WG envisions in the near time that MANETs will only be used as stub networks with all traffic beginning or ending within the MANET. This assumption restricts the usage of MANETs as a transit network for fixed or structured networks. The rationale for this restriction is based on expected bandwidth and power limitations of MANET nodes. [COR1] This constraint also reduces routing issues when a MANET is connected to the Internet allowing a mobile IP solution to be relatively effective. However as we will see later in our military operational example, MANETs may not only have to behave as transit networks for fixed elements, but multiples of MANETs will interact. Numerous protocols have been developed for isolated MANETs, which are a "special case" of their desired operation as part of a larger MANET. [BAK] Robust MANET routing protocols that support the most challenging implementations must also address nodal migration amongst MANETs and MANET migration across the Internet.

### **2.3 Characteristics of MANETS**

Scalability – MANETs range in scope from that of very low power sensors in an area of meters to WLAN extensions in a conference room or up to larger networks with hundreds of nodes as seen with military applications [COR2]. However, most MANET implementations do not scale well due to power, bandwidth and routing limitations.

Control – Wired networks and infrastructure based wireless networks like cellular use centralized control methods. Network control and management tasks are often consolidated at base stations or switches where processing power is abundant. Without this infrastructure MANETs employ a distributed control methodology where network decisions are made at each node for the benefit, or detriment, of the entire network.

Dynamic topology – MANET nodes arbitrarily enter, exit and move around the network architecture. This mobility can cause rapid, near random changes in the

network topology. Adjustments to transmission power and reception parameters, as well as the impact of terrain and structures also impact the topology [COR2].

Routing complexity – Routing in wired networks is a complex affair. Fixed routers have to contend with changing IP addresses, subnets, link-status and congestion control. In MANETs, since every node is a mobile router the routing complexity drastically increases. The dynamic topology causes constant changes to routing tables or source routes.

Bandwidth constraints and variable link capacity – Due to spectrum constraints and the nature of the medium, wireless networks have significantly less capacity than their wired counterparts do. Another effect of the medium is the asymmetry of link performance. The **changing wireless environment can lead to variable bandwidth-delay characteristics** [COR2] in the inter-nodal links. These facts, **coupled with potentially demanding applications, can cause increased congestion.**

Coverage – Coverage of a wired network is only limited by your ability to lay wire or cable. If there is a cable/wire infrastructure in place within 100m to a few kilometers (medium dependent) of your desired location then you are within coverage. Wireless networks have the advantage that you don't require a cable for every host, and you can rapidly build a network where there is no wired infrastructure. The addition of nodes to the network dynamically expands the network's coverage. This flexibility comes with drawbacks and costs. Wireless coverage is subject to the characteristics of the medium and the physical layer. It is also affected by environmental conditions, which can make the coverage inconsistent.

Energy constraints – Since MANET nodes are often powered by batteries power efficiency is a primary concern [COR2]. Every effort must be made to minimize needless transmissions and processing. This is difficult considering the additional tasks required for distributed control.

Reliability – Just as with coverage, reliability suffers from the effects of the medium. Topology changes, variable link capacity, power and energy issues affect reliability of the wireless network. Routing protocols must contend with these issues to preserve reliable delivery of messages.

Security – Communication systems are inherently vulnerable to security threats. Both wired and wireless networks are at risk to confidentiality and denial of service attacks. However, in many cases physical security measures are all that is necessary to secure a wired network. The wireless medium, on the other hand, is especially vulnerable and cannot be secured simply by physical security means.

## 2.4 Functions of MANET routing protocols

In the previous section, we discussed several characteristics of a wireless network that are not found in a wired network or in some cases not even required in an infrastructure based wireless network. Other characteristics discussed differ from wired networks in complexity. Developers and users of MANET routing protocols must consider these characteristics when designing and evaluating the protocols.

There are also several functions to consider that are affected by the characteristics previously discussed. The primary function of a routing protocol is to forward packets toward its destination in an efficient manner. In MANETs, a fundamental part of that function is topology/route discovery and maintenance. If a router (node) is to forward packets, it must know either the network topology or the route to the destination. Due the bandwidth and power, constrained environment the routing protocol must also consider resource efficiency. Finally, routing protocols must interact with adjacent layers in ways not typically found in wired networks. We discuss these functions in more detail in the remainder of this section.

### 2.4.1 Topology/route discovery and maintenance

Routing algorithms have two approaches regarding topology and routing information, non-adaptive and adaptive [COR1]. In non-adaptive protocols, the choice of the route to the destination is determined in advance and is downloaded to the routers when the network is booted. The adaptive protocols change their routing decisions reflecting changes in the topology. Adaptive algorithms differ in where they get their information, when they change the routes and what metrics are used for optimization.

MANET routing protocols must be adaptive, meaning that they can't rely on manual changes of routing tables that are initialized at network startup. Adaptive protocols are further defined as either proactive or *table-driven*, constantly updating routing tables thereby maintaining a global view of the network topology, or reactive, finding paths from source to destination *on demand* [COR1]. The best choice is often dependent on the rate of topology changes experienced by the network. Reactive protocols tend to be more efficient in networks with high rates of change.

A related task that is more difficult to implement in MANET routing protocols is loop prevention. Loops are created when topology changes occur between table updates or after a route has been determined. Rapid discovery of routing loops is essential since they are wasteful of resources and increase congestion in the network

### 2.4.2 Economy of resources

Routing protocols are often compared using factors such as routing overhead, packet delivery ratios, throughput and delay. Successful protocols often strike a balance between these metrics minimizing cost yet providing sufficient performance. In wired networks, there is more flexibility since over-provisioning of resources to improve performance is not so costly. In MANETs, the balance shifts toward economy of resources. Spectrum, power, link capacity, size and weight of devices are often bounded. MANET protocols should limit routing overhead, choose efficient routes, conserve bandwidth, and effectively manage power resources.

### 2.4.3 Interaction with adjacent layers

A typical wired network protocol may use **information about congestion or delay gained from other layers in its routing decisions**. This type of interaction is **even more important in wireless networks**. MANET protocols must ensure strong connectivity between nodes, be aware of neighboring terminals moving into its coverage range and learn ways to save network resources. Effective interaction with adjacent layers and implementing mechanisms that use information gained from these layers can improve overall network performance.

### **3 Operational Environment and Factors**

As stated earlier, the application and environment of a MANET has an impact on the effectiveness of the routing protocol used. As we developed our protocol hierarchy, we investigated scenarios in which the MANET would be used. We found that the current and conceptual military MANET implementations are overall the most difficult. From the number of nodes, topologies, and mobility to the physical environment, the potential military application of MANETs would challenge or overwhelm the best MANET protocols proposed to date. However, one advantage the military applications have over potential commercial ones is that they have management authority, administrative and technical control over all of the mobile nodes and most if not all of the fixed network interfaces. This has benefits in the realm of network administration, management and security. It also eliminates the problem of non-participating nodes that take advantage of network resources without contributing to the routing workload.

From this initial investigation, we pursued some broad organizational requirements that would clarify the scope of our network. In general terms, we wanted to determine the extent and relationship of MANET use, the geographic size and dispersion of the operational area and the quantity of nodes that would need to be supported. The following subsections describe these requirements.

#### **3.1 Organizational type and structure**

There are many different types of military organizations and applications that would benefit from MANETs. Applications range from scatterable sensor networks to logistics and transportation tracking systems and from small long-range reconnaissance teams to major headquarter locations that would want to limit cable infrastructure. These types of applications may have one or more demanding characteristics but the overall network complexity is tempered by another nodal aspect. For example, a scatterable sensor network may cover a large area using low power transmitters but it is relatively dense and is static once deployed. Whereas a transportation tracking system may be highly mobile with a sparse concentration of nodes but the platform is able to support high-powered transmitters. The most demanding military applications of MANETs are arguably the forward deployed tactical Brigades and Divisions. High mobility, low power transmitters and sparse or irregular dispersion challenge MANET implementations in combat formations.

#### **3.2 Unit Size and Corresponding Node Quantities**

We have chosen to further examine tactical combat units as our template but even these applications come in various types and configurations that can effect MANET implementation. A tank battalion can cover a much greater area than a light infantry battalion because of its platform mobility but it will have much

fewer nodes. A tank battalion has 58 tanks and an equivalent number of support vehicles [US1] of which all could support a MANET node. A light infantry battalion, on the other hand, has upwards to 500 soldiers who could be carrying a MANET node with few vehicles for mobility.

Since we wanted to assume a potential worst case for our network template we further focused our approach to consider mechanized infantry units, or the newly developed Stryker Brigade, both of which incorporate infantry fighting vehicles used for increased mobility, range and lethality in support of foot mobile infantry soldiers.

Using this general organization type, we developed the estimates found in Table 3-1 [US2][US3][US4] By examining the Battalion row we see that there are typically three to four battalions per brigade with each battalion supporting up to 500 MANET nodes in a 20 by 50 kilometer area.

| Organization | Quantity/next higher unit | Nodes per unit | Geography (km) |
|--------------|---------------------------|----------------|----------------|
| Squad        | 3-4                       | 9              | 1 x 1          |
| Platoon      | 3-4                       | 40             | 5 x 5          |
| Company      | 3-4                       | 150            | 10 x 10        |
| Battalion    | 3-4                       | 500            | 20 x 50        |
| Brigade      | 2-3                       | 2000           | 40 x 100       |
| Division     | N/A                       | 5000           | 120 x 200      |

**Table 3-1 Node Quantities and Geographic Area by Unit Type**

### **3.3 Battlespace Dimensions and Expected Nodal Dispersion**

In addition to node quantities, Table 3-1 also provides the geographical dimensions or battlespace in which each unit can be expected to operate. Doctrinal geographic area of employment estimates are available for division and brigade size units but vary dependent upon factors related to mission, enemy, terrain, troops and time. Estimates for battalion and smaller organizations are not found in current doctrine. Therefore, we developed estimates for these units based on higher unit estimates, organization structure and possible employment tactics.

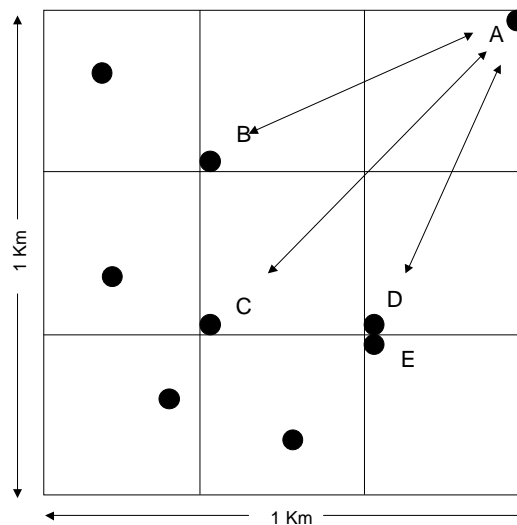
As we look at Table 3-1 it is obvious that the product of the geographic area with the quantity per next higher unit column is not equal to the geographic area listed for the next higher unit. It is important to note that each unit type is not required to occupy or “cover” the entire geographic area or that individual nodes must be equally dispersed across this area. The estimates merely suggest the largest area in which one would expect to find elements of the same unit operating.

As we move up in organization level, it can be shown that the area per node increases. There are several reasons for this that include increases in number of vehicles at higher echelons, an increasing number of support troops which occupy a portion of the area and the presence of greater fire support



systems. The predominant reason, however, is that there are large areas in which troops are not present. This could be because of mission decisions or that the terrain is impassable, i.e. mountainous or swampy, and is impossible or impractical to occupy.

Nodal dispersion is an illusive characteristic and is difficult to quantify. From the squad values in Table 3-1 we see that nine nodes operate in at most a square kilometer. Although unlikely, if we were to equally distribute the area amongst these soldiers then each soldier would be responsible for a 330m x 330m section of ground (Figure 3-1). If the soldiers had free reign to move about these areas of responsibility then it can be shown that the distance between any two nodes can be as great as 940 meters (vector AC) while the furthest distance a node can be from any other node can be as great as 740 meters (vectors AB or AD). Yet, at the same time it is possible for two nodes to be adjacent at the same point (nodes D and E). This simple illustration shows that even if we enforce nodal dispersion by grid assignment we still have huge disparity in transmission ranges.



**Figure 3-1 Squad Level Node Dispersion**

A more likely scenario is that nodes will form clusters with several nodes in very close proximity (within a few hundred meters). These clusters will be within kilometers of similar clusters forming cluster groups. Much longer distances, on the order of tens of kilometers, separate cluster groups. However, these clusters do not preclude migration between clusters, the appearance of isolated nodes, or a linear distribution “string” of nodes i.e. along main supply routes.

These facts are of interest to us as we develop MANET solutions because dispersion, density and physical obstructions play an important role in MANET protocol effectiveness and performance.

### **3.4 Target Unit Level**

The question that now arises is at what organization level do we want focus our MANET development. It is theoretically possible to consider every squad as a separate MANET as it is to say all mobile hosts in a division network belong to one MANET. The performance of either of these approaches would likely suffer due either to the many MANETs to be managed or the high number of nodes per MANET. The optimum partition is unknown at this point, but based purely on existing MANET protocol performance and the factors identified in the previous sections, a MANET at battalion level is likely the largest possible solution while a company MANET may be the smallest to consider.

A battalion supports approximately 500 nodes, which is the point of failure for many existing MANET protocols. Its area of operation is 20 x 50 kilometers which when coupled with an expected node transmitter range of 3-20 kilometers could easily result in more than six hops required for a message to traverse the MANET. It is also reasonable to assume that in any type combat formation used both the company and battalion level units would have the platforms necessary to support the potential use of high powered "super nodes" or "cluster heads".

### **3.5 IP Address Assignment Strategies**

In addition to developing a template organization, we also needed to research how the Army approaches IP addressing. We contacted the US Army Signal Center's Directorate of Combat Developments located at Fort Gordon, GA. The Army maintains two primary intranetworks, NIPRNET and SIPRNET. NIPRNET is the unclassified -but-sensitive (formerly Non-classified) Internet Protocol Routing Network that is primarily used in garrison networks. NIPRNET maintains minimally protected connections to the Internet. SIPRNET is the secret version and has both garrison and tactical components. Tactical units receive IP address assignments from a pool of forty Class B addresses assigned to the SIPRNET. Each Army Corps receives five SIPRNET Class B addresses and each Army element of a Unified Command such as US Army Europe receives four. The Army has only two NIPRNET class B addresses blocked to corps and commands.

IP address assignment strategies are different for each corps. XVIII Airborne Corps assigns their addresses using a relative hierarchy. Each of the XVIII Corps' divisions draw addresses from a single Class B and maintain management authority of those addresses. A division however does not receive a full Class B address. Excess address space is used for corps support units. The rationale for the hierarchical approach is primarily for management and control of the network. For instance, network managers use the Border Gateway Protocol (BGP) between corps and division and Open Shortest Path First Protocol (OSPF) internal to the division and corps networks. Providing a

controlled set of IP addresses allowed the Div manager to summarize their routers they control.

This raises the question of how reorganization is managed as in the frequent occurrence of attaching sub units from one organization to another. For example, a brigade from the 82d Airborne Division, XVIII Corps is attached to a division from the European based V Corps. Does the brigade keep their address space assigned from the XVIII or are they assigned new addresses from the V Corps address space? Although it can be handled many ways in most cases if they are moving into a completely different corps the unit obtains addresses from the receiving corps prior to deployment. If it were a reconstitution after battle, the unit would keep its existing addresses and use a BGP interface.

## **4 Wired Network Techniques**

We have discussed functions and characteristics of the MANET environment in general and we specifically targeted an operational environment, which we want to support. Before we delve into our solution, we should briefly discuss the wired network and the techniques used for networking and internetworking. Although the wired and wireless media are inherently different, there is something to be learned, and potentially carried over, from networking techniques used on the wired medium. In this section, we will explore local area networking, bridging, switching and routing and identify some of the techniques from which we developed our solution.

### **4.1 Local Area Networking (LAN)**

“A LAN is distinguished from other data networks in that it is optimized for a moderate sized geographic area such as a building or campus. A LAN is a shared medium peer-to-peer network that broadcasts information for all stations to receive.”[IEE] LANs operate at much greater speeds than other data network types due to their relatively short distances and medium access control (MAC) techniques. Ethernet is a popular example of a LAN standard which defines the physical and MAC layers. The MAC technique for Ethernet is Carrier sense multiple access with collision detection (CSMA/CD). This MAC technique allows the end stations (ES) of a LAN to share the same medium and avoid/recover from collisions without specifically dividing access in time. As collisions are inevitable in this type of shared medium the LAN segment is often called a collision domain.

LAN addressing schemes typically use device addresses that are unique, permanently assigned and are encoded into the network interface device. When an ES wishes to communicate with another end station it simply transmits the message with the recipients device (MAC) address included in the header. All end stations listen to the medium for their device address and copies the message only when its address is heard.

### **4.2 Internetworking**

In most cases, LAN users have a need to communicate or exchange data with individuals or systems beyond the limit of their LAN. There may be a need to link many similar or dissimilar network types in an organization or a requirement to link end users or LANs at geographically separate locations using a wide area network (WAN). A WAN is a network of multiple interconnected networks typically employing a backbone communication system. A WAN can be a private network owned and operated by a single organization or a public network that provides services to many organizations or to the public at large.

Internetworking is accomplished through a series of intermediate systems (IS) that may consist of bridges, switches or routers or a combination of the

three. Although a particular application may have the option of which IS to use each IS cannot support every application. For example, one could employ a router in place of a bridge but a bridge can't support every routing application. The choice of IS is dependent upon technical, topological and security requirements of the internetwork. [STA]

### **4.3 Bridging**

A bridge is the simplest of the three intermediate systems we will discuss and subsequently is the easiest to implement. A bridge, however, is limited in its capability. A bridge is a layer-two device used to interconnect two or more LAN segments that use the same LAN protocols at the MAC layer. [STA] A bridge is therefore only concerned with protocols up to layer two. Each LAN segment is assigned a physical port on the bridge. The bridge acts a packet filter by picking up packets heard on one LAN segment that are intended for an ES on another segment and retransmits them; it is in essence a MAC level relay. [STA] A bridge maintains a table for each segment that lists all device addresses heard on that segment. The bridge does not modify the contents of the packet nor does it add anything to packet. [STA] It simply reads the Ethernet addresses of all frames transmitted on every segment, checks them against the table, and repeats them if necessary on the appropriate segment. One of the greatest benefits of bridging is that to the end station, the bridged segments appear to be only one large LAN but by using multiple segments, you can allow multiple collision domains to coexist. This fact becomes very important as we examine wireless applications.

### **4.4 LAN Switching**

A LAN switch, like the bridge, operates at layer two and forwards packets based on device addresses. A switch also does not modify the contents of the packet nor does it add anything to packet. A switch is used in Star Topology LANs in which each end station is assigned a unique port. Although a Star LAN is not a physical bus, it is logically one. The switch monitors every port and checks incoming packets' source addresses. It maintains a table that specifies from which port it hears a particular device address. When an ES transmits a packet destined for a particular device address the switch checks its table and determines the location (port) the destination is located and retransmits. In a switch, the forwarding function is performed in hardware, which enables the switch to achieve higher performance.

Switches can be placed in tandem simply by assigning another switch to a port. The switch tables must then account for a range of device addresses that will be heard on that port.

## 4.5 Routing

A router is a complex general-purpose device that can be used to connect dissimilar networks. It operates at layer three of the OSI model and therefore need only have compatible protocols up to the network layer. The router must be able to contend with many different network characteristics that include address schemes, frame sizes, interfaces and network services. [STA]

The routers operation depends on which network layer protocol is used. If IP is used then IP must be implemented on all ends stations, the LAN and in all routers. If a station on a LAN wants to send a packet to a station on another network connected by one or more routers then the higher layer protocols pass the frame down to the IP layer. IP then adds the destination's global IP address and determines if the destination is on a different network. It does this by examining the network ID portion of the IP address and compares it to its own network ID. If different, then the message must be sent to a router that can move the packet closer to its destination. IP then passes the frame down to the LLC layer identifying the router. LLC passes the frame to the MAC layer, which adds the MAC address of the router and places the frame on the LAN. When the router receives the frame, it strips off the MAC address, LLC header and examines the IP address in order to make an appropriate routing decision. If the router is directly connected to the destination LAN as specified by the IP network ID then the router will add a MAC header to the frame that includes the device address of the end station. [STA]

As demonstrated in the previous example a router must modify the packet by removing the MAC and LLC headers to examine the IP datagram. This time consuming process can add significant delay if a packet must traverse several routers. A mitigating factor to this delay is the table lookup process. In a wired network, the router is only concerned with the destination's network ID portion of the IP address and not the host ID. This drastically limits the number of table entries in which a router must maintain. In ad hoc wireless networks, the routing protocols will often perform host-to-host routing in which the node must maintain a table of all destination nodes. This is not a significant problem in small MANETs that represent a single sub domain but when a network contains large MANETs, MANETs that support nodes from multiple domains or multiple MANETs that migrate around the wired network then host-to-host routing can pose a serious problem.

## 4.6 IP over Ethernet

In the previous section, we discussed the basic operation of IP routing and how the IP protocol identifies to the lower layers the desired router to be used. IP does this by providing to the Link Layer the router's IP address. We must recall that local area networks do not use IP addresses but device addresses to broadcast frames on the medium. Therefore if the MAC address of the router is not known then a process must be initiated to discover it. IP uses a protocol that

resides below the network layer and above the MAC layer called the Address Resolution Protocol (ARP) to discover the destination MAC address. ARP broadcasts a query for the MAC address onto the network using the IP broadcast address (corresponds to MAC address of all 1's). Any routers connected to the network will not rebroadcast the request beyond the local network. All stations on the LAN, including routers, examine this query and decide if their IP address matches the one included in the query. If so, it responds identifying its MAC address to the source. If the destination is outside the local network, the ARP query will receive the MAC address of the router with a route toward the destination's network.

#### **4.7 Wireless adaptation**

In the previous sections, we discussed several aspects of networking using a wired medium. As we move to discuss wireless networks, we must consider that the environment has many distinct differences that go well beyond the physical network interface and medium characteristics. Routing, medium access and security decisions are all impacted when wireless networking is considered.

First let's consider a wireless LAN. We learned that an Ethernet LAN segment is considered a single collision domain with all stations hearing all others on the segment. In a wireless network, there is no guarantee that all stations are within transmission range or without an obstruction. Therefore, the collision domain for a wireless network is fragmented and the network view for each node is different. We also learned that bridges in a wired network connect like LAN segments thereby separating collision domains. However, we are unable to directly adapt this bridging concept in a wireless network using a common medium since there are no physical ports to delineate LAN segments. We could attempt to use frequency, space or code diversity to uniquely identify segments but the administration of such a scheme in a mobile environment would be prohibitive. The same principle applies for a switching implementation in a wireless network. There are no physical ports to delineate destinations for switching.

The accepted answer to this dilemma in current ad hoc networking literature is to use pure routing. The IETF MANET WG members have come to the consensus that the approach to ad hoc networking should be network layer routing with each node acting as a router. However, we also discussed in earlier sections that host-to-host routing does not scale and becomes very complex in our operational environment.

In the following section, we will introduce our approach called Virtual MAC Tag Switching. VMTS takes its inspiration from MAC layer bridging. VMTS uses virtual bridge architecture with each node potentially performing bridging functions. VMTS is not pure bridging since we modify the packet at each bridge by adding a TAG that includes next hop MAC address, hop count, lifetime and possibly path information. VMTS is not switching since a node can't listen to

individual ports. VMTS does listen and learn whom its neighbors are and maintains a switch-like table where instead of entering the port to the neighbor it enters the next hop. VMTS is not IP routing since it routes using layer two MAC addresses not IP addresses but it exhibits many characteristics of ad hoc routing protocols.



## 5 VMTS protocol description

In this section, we describe the Virtual MAC Tag Switching (VMTS) protocol. We should note that the algorithm is presented in its final form, which was influenced by implementation issues that arose during the coding of the protocol for simulation in ns and has evolved from our initial design proposal.

Throughout this section, we will use examples to explain the protocol. To illustrate these examples we will use the scenario represented in the node link connectivity diagram shown in Figure 5.1. It is an example of unit areas of coverage, boundaries, and potential nodal connectivity. From left to right, the unit areas represent those of 1<sup>st</sup> and 2<sup>nd</sup> Battalion of the 1<sup>st</sup> Brigade and 1<sup>st</sup> Battalion of the adjacent 2<sup>nd</sup> Brigade. Within the boundaries are various lettered nodes and an external connection to the Internet or Intranet. As pointed out in Table 3-1 in the operational factors section, the number of nodes within a particular unit sector would be significantly higher than what is represented here. However, this example provides a fair representation of the types of connectivity and routing challenges one would face in a tactical military MANET.

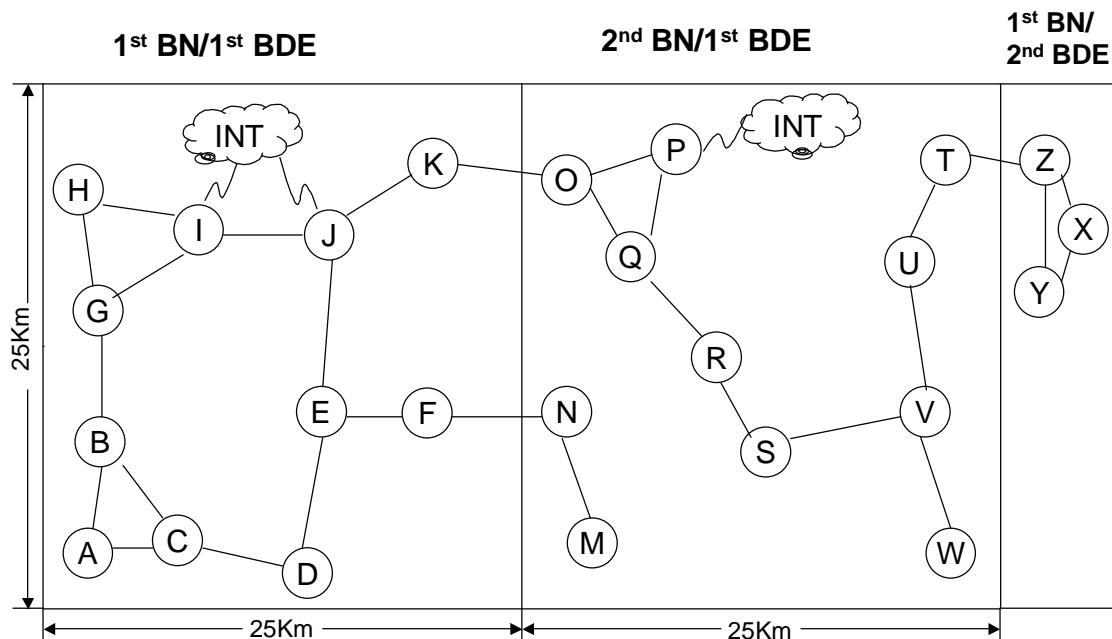
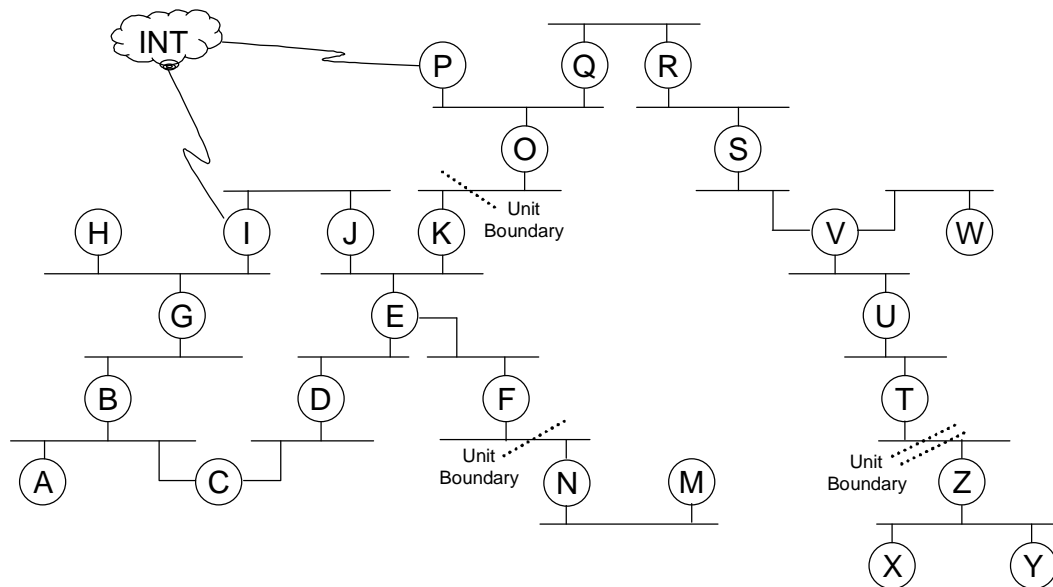


Figure 5-1 Node Link Connectivity Diagram

Figure 5-2 shows the equivalent bridge network topology that represents the previous node connectivity diagram. It shows how a node may be simply an endpoint on a virtual LAN segment or may need to perform bridging duties. It is clear that this network is not loop free and since maximum link connectivity in MANETs is desired the bridging/routing techniques must contend with loops in the network topology.



**Figure 5-2 Node Bridge Topology Diagram**

## 5.1 General characteristics

VMTS is a Hop by hop distance vector protocol operating at the MAC layer. VMTS was inspired by MAC bridging and therefore routes using MAC addresses supplemented by a MAC\_Tag that adapts the protocol to the wireless environment. To reduce overhead and improve scalability of VMTS, we have eliminated prescribed control or routing messages between nodes. There are neither hello messages nor periodic routing updates and subsequently, VMTS does not perform partial or complete table updates as seen in most table driven ad hoc protocols. Neighbor and path discovery is accomplished through passive measures and active mechanisms incorporated into the VMTS MAC\_Tag.

VMTS requires four MAC addresses for each message. The additional addresses can be applied to the MAC\_Tag or incorporated into the MAC header using the four-address option of the 802.11b protocol. This option affords four addresses: receive (next hop), transmit (source or forwarding node), destination and source in the MAC header. We chose the former to accommodate the simulation environment that we will discuss in section six.

The VMTS routing agent attached to each node in the MANET performs all routing decisions for messages intended for nodes within the scope of the MANET. The routing agent maintains a table of destinations that identifies the next hop toward the destination. The table also keeps a distance metric consisting of the number of hops to the destination and a decrementing lifetime value for the entry. The following sections describe in detail the operation of the VMTS routing agent, the routing table and other mechanisms useful to the performance of the protocol.

## 5.2 VMTS routing

Routing in VMTS is accomplished by five mechanisms: route discovery, message forwarding, route repair, neighbor re-associate and neighbor discovery. These mechanisms and other supporting features are further described in this subsection and are included in Figure 5-3.

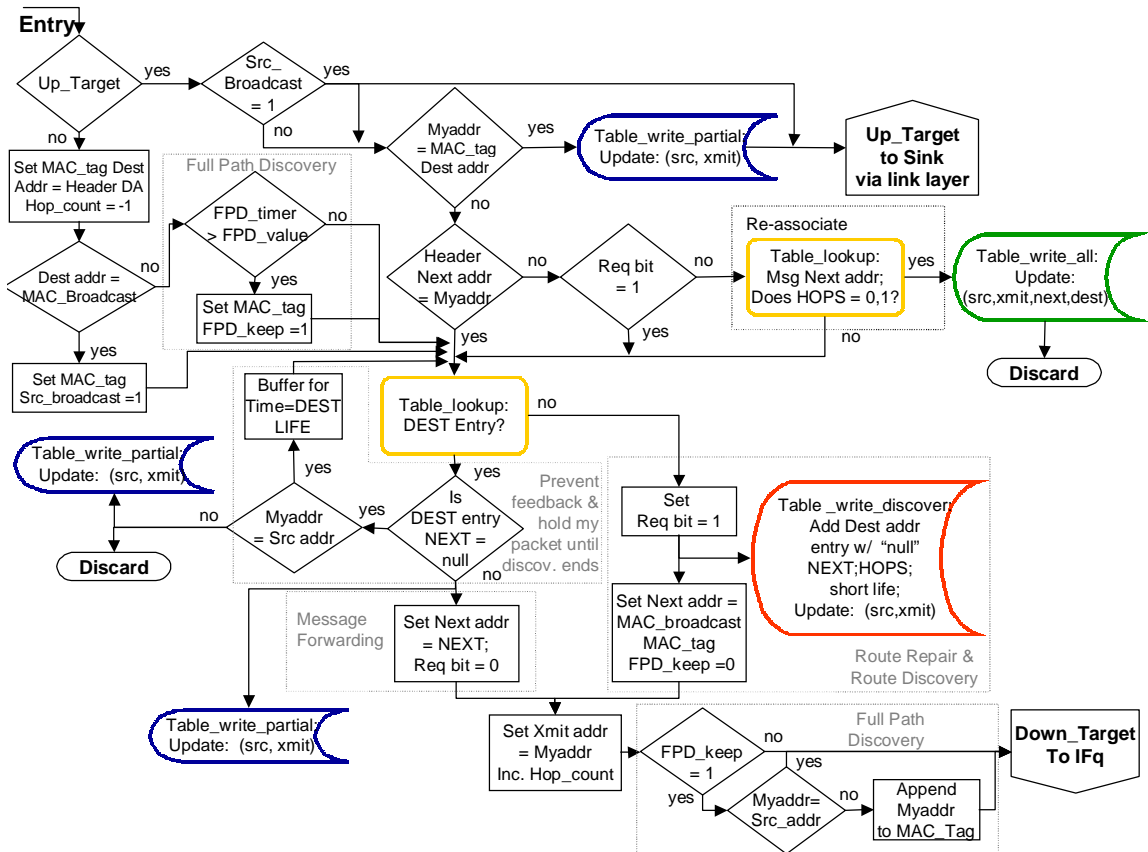


Figure 5-3 VMTS Routing agent process

### 5.2.1 Route discovery

When a source on a node has data to send but does not have an entry in its table for the destination, it implements a route discovery process. Route discovery is not a formal route request message as found in many ad hoc routing protocols but is achieved by merely indicating a request bit in a packet's MAC\_Tag. To find a route to a destination, a node transmits a message with the request bit indicated (bit=1), the MAC broadcast address assigned to the MAC Header next hop address and the destination address set to the message final destination. When a neighbor node hears the message on the channel, it passes it to the routing agent, which checks to see if it has a valid next hop to the message destination. If the agent has a next hop for the destination it resets the

request bit to zero (thereby limiting broadcasting), increments the hop count metric, adds the next hop toward the destination from their table to the next hop address field of the MAC header, adds its own address to the transmitting address field of the MAC\_Tag and forwards the packet to the next hop.

If the neighbor node doesn't have a valid entry for the destination node, the neighbor node continues the broadcast by incrementing the hop count, adds its own address to the transmitting address field of the MAC\_Tag and returning the packet to the channel.

To prevent recursive broadcasting (ping-pong effect) of the same discovery message, when a node forwards the message it makes a temporary "place holder" entry in its table. This entry includes the packet's destination and a short lifetime value. The remaining entry fields are given a "null" value. If the forwarding node hears other nodes transmitting (retransmitting) discovery messages for the same destination, it checks the table, sees the "null" next hop entry and does not forward the message. If a route discovery is in progress for a destination and a source has traffic for the destination, it will delay sending its traffic until the end of the discovery period.

### **5.2.2 Message forwarding**

Message forwarding occurs when a node is identified as the next hop toward the message destination. A forwarding node hears the message on the channel and sees that its address is the message's next hop address. The forwarding node routing agent increments the message's hop count metric, changes the next hop address to the next hop listed in its own table, adds its own address in the transmitting address field and passes the message down to the network interface to be placed on the channel.

As discussed previously in the route discovery subsection, if a node's routing agent receives a broadcast message but knows the path (next hop) to the destination it ends the broadcast by resetting the request bit (bit=0) and proceeds with normal message forwarding.

### **5.2.3 Route repair**

Route repair is a mechanism that attempts to repair a path to the packet destination when it is discovered that the path is broken. If a node hears a message for which it is a forwarding node, but no longer has a valid entry for the message destination, it must take action to ensure the message continues on the path to the destination. The forwarding node repairs the route by initiating the discovery process for the message on behalf of the message source. The node's routing agent indicates the request bit (bit=1) of the message, increments the metric, sets the next hop address to the MAC broadcast address, adds its address to the transmitting address field and passes the message to the channel.

It is important to note that as the route discovery process begins, the message will propagate back toward the message source. This may cause

intermediate nodes to enter additional higher metric entries for the message source into their routing tables. This is not a concern because the accurate lower metric entries will take priority in future messages.

#### **5.2.4 Neighbor re-associate**

With the node migration that is characteristic of MANETs, a node may not realize it has moved out of range of its previous neighbors. In order for new or migrating nodes to immediately re-access the network and begin sending traffic, we implemented the neighbor re-associate mechanism in VMTS. When a node hears a message that does not address it as the next hop and the request bit is not set, the routing agent checks its table for the next hop address. If it finds that the next hop is not listed in its own table as a zero-hop or one-hop neighbor (HOPS in table =0,1), it determines that the source node has migrated away from its previous neighbors. The node assumes responsibility for the message and retransmits the message according to the applicable message forwarding or route repair processes. This process also informs the message source of its new neighbors since it will overhear its new neighbors transmissions.

#### **5.2.5 Neighbor discovery**

Since there are no hello messages, route requests or period table updates neighbor discovery becomes primarily passive. Neighbor discovery occurs as a result of “listening” to passing traffic and learning information about neighbors and destinations from message headers. Exactly how this occurs is discussed further in the Promiscuous Mode and the Routing Table Process sections.

#### **5.2.6 ARP requests and broadcast messages**

Since VMTS resides below the link layer in the protocol stack it must handle Address Resolution Protocol (ARP) requests and other broadcast packets from upper layers such as ICMP messages. We have added a Src\_broadcast bit in the MAC\_Tag to delineate source initiated broadcast messages from VMTS discovery messages. Upon receipt of an ARP request or other source broadcast message from the upper layers of the node's protocol stack, VMTS sets the Src\_broadcast bit. The ARP is then handled as any other VMTS broadcast message before it is placed on the channel. Nodes receiving an ARP packet check the Src\_broadcast bit and immediately send the packet up to the link layer for processing by upper layer protocols. VMTS also propagates the message to other nodes as is required for the proper handling of the message. Prior to passing the message back down to the channel, it processes the message header and MAC\_Tag learning new destinations and updating existing cache entries as applicable.

### 5.2.7 Full path discovery (FPD)

The passive neighbor discovery mechanisms in conjunction with the route discovery process provide an effective means of determining paths to destinations. However, as you will learn later in the table subsection, the only destinations that are learned are those that are on either end of the message path. In paths with a large number of hops, there may be several nodes through which the message travels that are not added to node tables.

VMTS includes an optimization whereas with some probability  $p$  a packet will capture information regarding the entire path that it travels, thereby capturing the path information to intermediate nodes. Each node along the message path appends its address in the MAC\_Tag creating a full path. The path information is used to further populate and update VMTS routing tables. We call this optimization full path discovery (FPD).

We chose to implement the FPD optimization in the routing agent process simply by using a variable linked to the node's clock. The FPD\_timer increments with time until it reaches an adjustable limit called FPD\_value. Once the limit is reached, the FPD\_timer is reset and the next message sent by the node's routing agent would enact the full path discovery. In practice, synchronization is not a concern since a node's start time is randomly distributed and the time from when FPD\_timer reaches its limit until a message is sent is random. However, if we start all nodes at the beginning of simulation time will see an increase of FPD packets at FPD\_timer intervals. Therefore, in simulation we must jitter the node start times to prevent synchronization.

### 5.3 Promiscuous mode

A key element of the VMTS protocol is the application of the promiscuous mode capabilities of the physical layer. With promiscuous mode operation, the node's channel interface passes every data packet heard up to the routing agent. The agent checks the MAC\_Tag and MAC header for addresses resulting in neighbor discovery, table maintenance and new destination information.

Information discovered from promiscuous mode operation includes:

- Learns next hops to new sources and destinations.
  - o Learns next hop to the data packet's source (source address is entered into node's table as the destination with the node from which the data packet was received entered as the next hop. The packet hop count is captured as the distance metric to the source)
  - o Learns next hop to the data packet's destination (destination address would be entered into table with the node from which the data packet was received entered as the next hop)
  - o It also learns the next hop to the node transmitting the packet (if different than the source) and the packets next hop (if different than the destination)

- However, in the case of a packet with the Request bit indicated the node would not capture the data packet destination information since the destination route is not yet established
- Learns all zero-hop neighbors during any broadcast message.
- Enables table updates for existing destinations when a more efficient route is heard or if less efficient but fresher route is heard.

## 5.4 Routing table

The routing agent for each node in the MANET manages a MAC level routing table populated with the MAC addresses of destinations and next hops. Also included are the hop count metric and a decrementing lifetime. An excerpt of Node A's routing table in our example network from Figure 5-1 is shown below in Figure 5-4.

| NODE A |      |      |          |
|--------|------|------|----------|
| DEST   | NEXT | HOPS | LIFE (s) |
| E      | C    | 2    | 10       |
| B      | B    | 0    | 15       |
| C      | C    | 0    | 6        |
| F      | null | null | 0.014    |
| E      | B    | 4    | 80       |

Figure 5-4 Node A routing table

### 5.4.1 Table fields

The fields of the routing table are further defined:

DEST – A destination node MAC address

NEXT – The MAC address of a node within transmission range of Node A (zero-hop neighbor) that Node A knows to be the next hop to the destination node – DEST.

HOPS – The number of hops to the destination as captured from the MAC\_Tag

LIFE – The time remaining in seconds for the table entry to be considered valid. Initially set to an adjustable value at the time the entry was made. The value of the LIFE field for each entry decrements with the simulation clock

### 5.4.2 Null fields

During the discovery process (MAC\_Tag request bit=1), the source and forwarding nodes add a table entry for the packet destination but assign a "null" value to the NEXT and HOPS fields. A short LIFE value is entered for this temporary entry. Upon receipt of the message, the destination routing agent enters into its table the various routes discovered. When the source acknowledges the message, it will choose the next hop for the path with the

lowest metric. The forwarding nodes along this route create a standard entry including NEXT, HOPS and standard LIFE values. The entry in nodes not along this route will expire after the short LIFE.

### 5.4.3 Table lookup procedure

The routing agent calls a table lookup procedure that returns a confirmation of an entry for a destination and information regarding the entry that is used in routing decisions. A flowchart for the procedure is shown in Figure 5-5. The agent for Node A performs a table lookup for every packet initiated from its node's sources with the exception of source broadcast packets. It also performs table lookups for every packet heard on the network interface except for packets destined for Node A or packets listing a next hop of one of Node A's zero or one-hop neighbors.

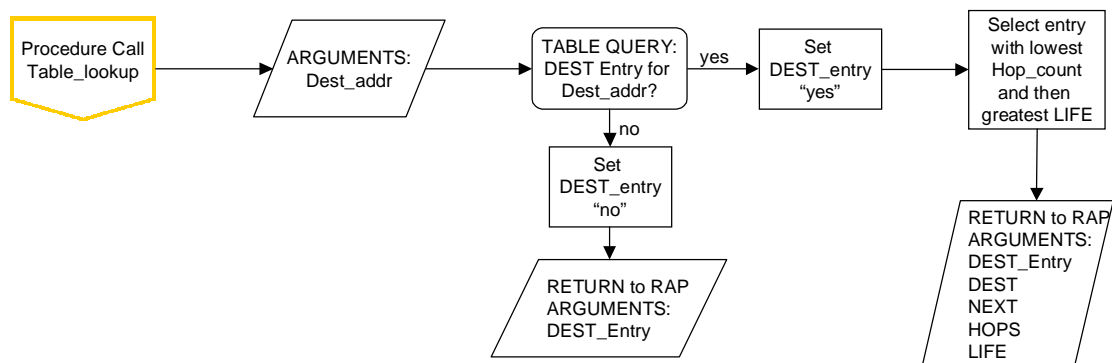


Figure 5-5 Table lookup procedure

### 5.4.4 Table write procedures

The routing agent adds entries as described in section 5.4.1 for nodes discovered from MAC\_Tags and MAC headers. This is the only method by which destinations are discovered and the tables are populated. From a single message the routing agent may add or update a DEST entry for:

- The message source indicating the transmitting node as the NEXT hop, the hop count listed in the message MAC\_Tag (Hop\_count) as the HOPS metric and new lifetime value for LIFE.
- The transmitting node (if different from the source) with the transmitting node as the NEXT hop, zero as the HOPS metric and a new LIFE.
- The message destination with the transmitting node as the NEXT hop, a maximum HOPS (32 since it is a 5-bit field) and a new LIFE. (Notes: We use the transmitting node as the next hop because the message's next hop may be beyond our range. If the next hop is within range then the entry will be corrected when that transmission is heard. We use a max

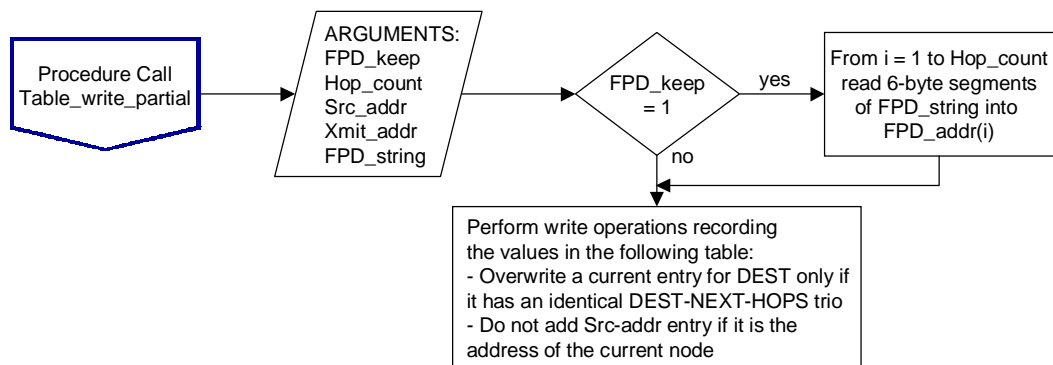


metric because we have no way of knowing the number of hops to the destination and we don't want to override a known shorter route.)

- The message next hop address with the transmitting node as the NEXT hop, a maximum HOPS (32 since it is a 5-bit field) and a new LIFE.
- The message destination with "null" NEXT, HOPS fields, and a short LIFE during a route discovery.
- All nodes indicated in the FPD Route Addresses field of the MAC\_Tag when *Full Path Discovery* is implemented. The transmitting node is entered as the NEXT hop for all; (Hop\_count - 1) is the HOPS metric for the first address in the field, Hop\_count-2 for the next and so on until all addresses are exhausted. A new LIFE is also entered for each entry.

The routing agent uses three unique write procedures that enter different combinations of the above records. The selection of the procedure depends on whether the message received is a forwarding message, route discovery, overheard on the channel or reached its final destination. All procedures will capture FPD addresses.

The first procedure is a partial write procedure. A flowchart of the procedure and the table entries made are shown in Figure 5-6. It is used when a message has reached its destination, when the routing agent is forwarding a message or when it receives broadcast messages from other nodes for a destination currently in route discovery.



Note: We have excluded the source address from FPD\_string since it is repetitive and wastes resources.

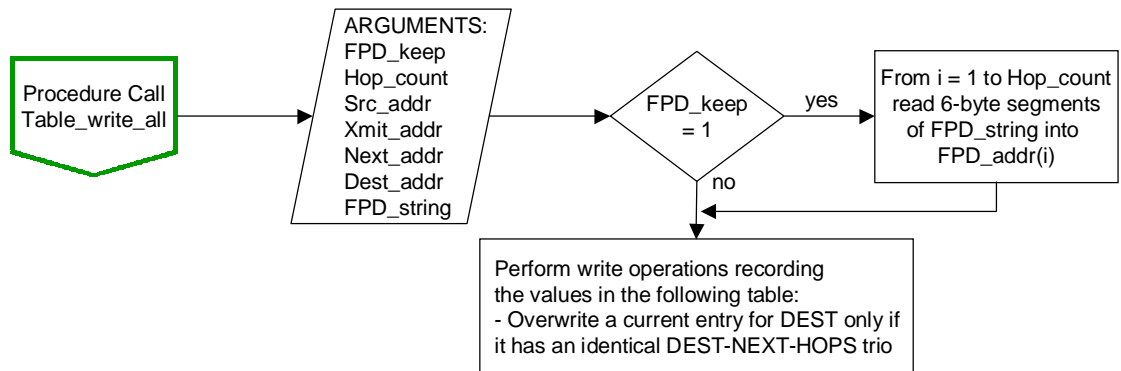
Also it is not necessary to enter into the table the FPD\_addr(Hop\_count) as it is the same value as the Xmit\_addr entry

| DEST                  | NEXT      | HOPS         | LIFE     |
|-----------------------|-----------|--------------|----------|
| Src_addr              | Xmit_addr | Hop_count    | New_life |
| Xmit_addr             | Xmit_addr | 0            | New_life |
| FPD_addr(1)           | Xmit_addr | Hop_count-1  | New_life |
| FPD_addr(2)           | Xmit_addr | Hop_count- 2 | New_life |
| ...                   | ...       | ...          | ...      |
| FPD_addr(Hop_count-1) | Xmit_addr | 1            | New_life |

**Figure 5-6 Partial table write procedure**

The routing agent may also call a full write procedure. A flowchart of this procedure is in Figure 5-7. The full write is used for all messages heard on the

network interface except broadcast messages that require no action from the routing agent.



| DEST                  | NEXT      | HOPS         | LIFE     |
|-----------------------|-----------|--------------|----------|
| Src_addr              | Xmit_addr | Hop_count    | New_life |
| Xmit_addr             | Xmit_addr | 0            | New_life |
| Next_addr             | Xmit_addr | 32 (Max)     | New_life |
| Dest_addr             | Xmit_addr | 32 (Max)     | New_life |
| FPD_addr(1)           | Xmit_addr | Hop_count-1  | New_life |
| FPD_addr(2)           | Xmit_addr | Hop_count- 2 | New_life |
| ...                   | ...       | ...          | ...      |
| FPD_addr(Hop_count-1) | Xmit_addr | 1            | New_life |

**Figure 5-7 Full table write procedure**

The final write procedure available is used during the route discovery process. As explained earlier, during a route discovery the source and broadcasting nodes enter a temporary entry to prevent recursive broadcasting of the same message. The discovery write procedure adds this temporary entry assigning a short lifetime. The flowchart for the discovery table write procedure is seen in Figure 5-8.

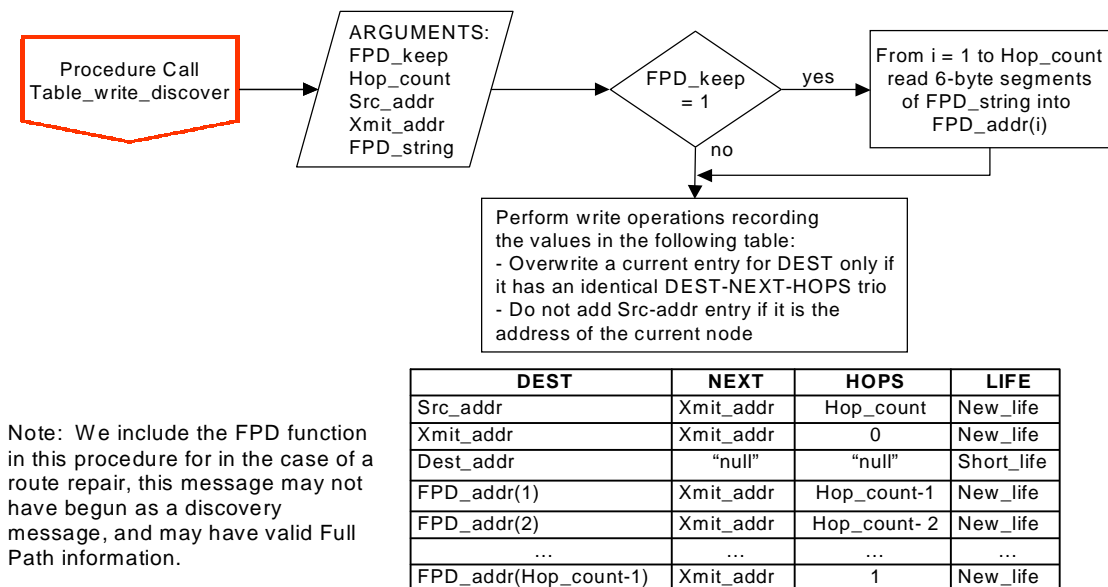


Figure 5-8 Route discovery table write procedure

### 5.4.5 Multiple entries per destination

Nodes may keep multiple routes for a destination but select the path with the least hop metric even though it may have a shorter lifetime (older route). If there is more than one route with equal metric then it uses the one with the longest LIFE. If the older (shorter) route is no longer valid, forwarding nodes will discover the faulty route, and will either repair the route or initiate a new discovery process. Although the procedure allows multiple entries for a destination, it must prevent duplicate entries as evaluated on the DEST, NEXT and HOPS fields.

### 5.5 VMTS MAC Frame

According to the IEEE 802.11-1997 standard specification, the generic format for the 802.11 MAC frame includes a header consisting of a frame control field, ID/Duration field and up to four address fields. The frame also includes a tail that performs a CRC on the header and link layer PDU. The format for the generic Frame is shown in Figure 5-9. The four address fields include source and destination addresses with the option for two other addresses such access points or transmitted and received addresses.

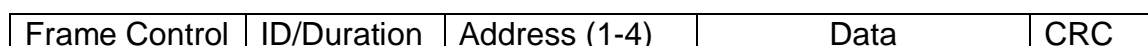


Figure 5-9 Generic 802.11 MAC frame

The network simulator implementation of the 802.11 MAC layer only uses two of these addresses, the source and destination address. Since we needed four addresses to accommodate the transmitting and receiving next hop nodes, we opted to include two addresses in the MAC\_Tag. We maintain the source address in the MAC header but move the destination address to the MAC\_Tag. We added the next hop address field to the MAC header and the transmitting address to the MAC\_Tag. The resulting MAC Frame is shown in Figure 5-10.

|               |             |                |                  |      |     |
|---------------|-------------|----------------|------------------|------|-----|
| Frame Control | ID/Duration | Source Address | Next Hop Address | Data | CRC |
|---------------|-------------|----------------|------------------|------|-----|

Figure 5-10 VMTS MAC frame

### 5.5.1 MAC\_TAG

The VMTS protocol adds a MAC\_Tag to the MAC header to contend with several issues that arise when you use MAC level routing in a common channel wireless environment. The MAC\_Tag format is given in Figure-11. The MAC\_Tag is a minimum of thirteen bytes long and can increase in length by the product of the Hop\_count value and the length of a MAC address (six bytes) when full path discovery is used. The minimum MAC\_Tag includes a one-byte control field and two six-byte address fields.

|             |               |            |           |                      |                     |
|-------------|---------------|------------|-----------|----------------------|---------------------|
| Byte 0      | Bytes 1 => 12 |            |           |                      |                     |
| Bit 0       | 1             | 2          | 3         |                      |                     |
| Request Bit | Src Broadcast | Keep Route | Hop_Count | Transmitting Address | Destination Address |

Bytes 13 => [13 + (Hop\_count x 6)]

|  |
|--|
| Full Path Discovery Addresses (1 => Hop_count) |
|--|

Figure 5-11 VMTS MAC\_Tag

The VMTS control field is comprised of three indicator bits and a five-bit hop count. These values are:

Request Bit (1 bit) - Used if next hop to destination is not known as in a route discovery or broadcast packet. A value of 1= destination search requested.

Source Broadcast (1 bit) - Used to identify ARP request and other upper layer generated broadcast packets

Hop Count (5 bits) - Incremented by each forwarding node. Used as a distance metric in routing tables. Also used to determine length of MAC Tag if Full Path Discovery is used.

Keep Route (1 bit) - Used with Full Path Discovery option. If indicated (bit=1) each forwarding node appends its address at the end of the

header. Nodes hearing the packet on their interfaces may use the path information to update their routing tables. Since the source node address is carried in the MAC header, the VMTS routing agent does not add its address to the FPD path.

FPO Route Addresses (6 bytes x Hop\_count) - See Keep Route. Length of field depends on the number of hops the packet has traveled.

## 5.6 VMTS protocol routing example

In this section we will demonstrate VMTS through a simple multipart example. Throughout the example, we have included snapshots of the relevant nodes' routing tables. For simplicity, we have removed reference to the entry lifetimes. Consider our operational example from Figure 5-1 and the equivalent bridging diagram from Figure 5-2 at network startup when all tables are empty:

Part I: Node A wants to send node E a message.

- Node A checks its table for node E and doesn't find an entry
- Node A broadcasts the message using the MAC broadcast address with E's destination address in the MAC\_Tag
- Nodes B and C hear this message on their interface, check and find that they have no entry for the destination E in their tables.
- They enter the source node A into their tables and include a hop count of 0 indicating that it is a zero-hop neighbor (some authors consider this a 1-hop neighbor but in the context of MAC layer routing these nodes are on the same virtual LAN segment, therefore no hops from segment to segment are necessary to reach them).

| NODE B |      |      | NODE C |      |      |
|--------|------|------|--------|------|------|
| DEST   | NEXT | HOPS | DEST   | NEXT | HOPS |
| A      | A    | 0    | A      | A    | 0    |

- Nodes B and C rebroadcast the message on the same interface incrementing the hop count.
- Node A hears the rebroadcast message and updates its table by adding nodes B and C as 0-hop neighbors. Nodes G and D also hear the message, update their table with both the message source and the forwarding (transmitting) node from which they received the message. Node G and D check their table and rebroadcast the message.
- In order to prevent a source or node from retransmitting a message heard on its interface that it has already transmitted the node enters a temporary entry for the message destination with null next and hop count fields.

| NODE A |      |      | NODE D |      |      | NODE G |      |      |
|--------|------|------|--------|------|------|--------|------|------|
| DEST   | NEXT | HOPS | DEST   | NEXT | HOPS | DEST   | NEXT | HOPS |
| E      | null | null | A      | C    | 1    | A      | B    | 1    |
| B      | B    | 0    | C      | C    | 0    | B      | B    | 0    |
| C      | C    | 0    | E      | null | null | E      | null | null |

| NODE B |      |      | NODE C |      |      |
|--------|------|------|--------|------|------|
| DEST   | NEXT | HOPS | DEST   | NEXT | HOPS |
| A      | A    | 0    | A      | A    | 0    |
| E      | null | null | E      | null | null |
| C      | C    | 0    | B      | B    | 0    |
| G      | G    | 0    | D      | D    | 0    |

- Nodes H, E and we hear the broadcast message and update accordingly. Both H and I rebroadcast even though H is not currently acting as a bridge. It does this for several reasons that include neighbor updates and discovery of new nodes that may have entered H's transmission range. This could generate collisions but is something that the physical layer must contend with through collision avoidance in order to prevent isolation of nodes.

| NODE H |      |      | NODE I |      |      | NODE E |      |      |
|--------|------|------|--------|------|------|--------|------|------|
| DEST   | NEXT | HOPS | DEST   | NEXT | HOPS | DEST   | NEXT | HOPS |
| A      | G    | 2    | A      | G    | 2    | A      | D    | 2    |
| G      | G    | 0    | G      | G    | 0    | D      | D    | 0    |
| E      | null | null | E      | null | null |        |      |      |
| I      | I    | 0    | H      | H    | 0    |        |      |      |

| NODE G |      |      |
|--------|------|------|
| DEST   | NEXT | HOPS |
| A      | B    | 1    |
| B      | B    | 0    |
| E      | null | null |
| I      | I    | 0    |
| H      | H    | 0    |

- Node J hears node I's transmission, checks the table, updates and rebroadcasts. Note: had node E already acknowledged receipt of the packet from node D, then J would have heard E's transmission and entered E in his table. This would have kept J from broadcasting the message.
- Node E hears a message again from A, this time through J. Node K also hears the message and rebroadcasts. Node E decides that since route from D is shorter it will respond through node D. It will update its table with K and J's information and will keep the route through J as an alternate route.
- The nodes along the shorter route will update their tables with E's information as the return message makes it way to node A. (Node B will

also complete the entry for node E since it will have overheard the return message identifying the shortest path through node C).

| NODE D |      |      | NODE C |      |      | NODE E |      |      |
|--------|------|------|--------|------|------|--------|------|------|
| DEST   | NEXT | HOPS | DEST   | NEXT | HOPS | DEST   | NEXT | HOPS |
| A      | C    | 1    | A      | A    | 0    | A      | D    | 2    |
| C      | C    | 0    | D      | D    | 0    | D      | D    | 0    |
| E      | E    | 0    | B      | B    | 0    | J      | J    | 0    |
|        |      |      | E      | D    | 1    | K      | K    | 0    |

Part II: We now want node F to send a message to node C.

- The process above repeats with node F broadcasting a message using the route discovery process.
- Node E hears the message and rebroadcasts since node E is not aware of a route to node C. If a full path discovery had been used on the previous message, node E would have knowledge of the path to node C and would forward the message to node D only.
- Nodes J and D hear the discovery message from E. Node J rebroadcasts the message continuing the route discovery but node D stops the discovery process because it knows the path to C.
- Node D adds the correct next hop to the message and forwards it to node C.

Part III: Node A wants to send another message to node E.

- Node A transmits a message with node E as the destination and node C as the next hop, node C receives the message and forwards to node D and so on to node E.
- Node B also hears the transmission. Since it is not a broadcast, it does not immediately rebroadcast. It does however check its table to see if node C is a zero or one hop neighbor. If C weren't a neighbor, node B would retransmit the message re-associating the message source (node A) to its new neighbors. Since node C is a zero-hop neighbor, node B does not retransmit. Node B adds destination E to its table with C as the next hop and a max hop-count since it does not know the actual hop count to the destination.

| NODE B |      |      |
|--------|------|------|
| DEST   | NEXT | HOPS |
| A      | A    | 0    |
| C      | C    | 0    |
| G      | G    | 0    |
| E      | C    | 32   |

Part IV: Consider what occurs when node A leaves the transmission range of nodes B and C.

- From the previous examples the table entries for nodes A, B and C are:

| NODE A |      |      | NODE B |      |      | NODE C |      |      |
|--------|------|------|--------|------|------|--------|------|------|
| DEST   | NEXT | HOPS | DEST   | NEXT | HOPS | DEST   | NEXT | HOPS |
| E      | C    | 2    | A      | A    | 0    | A      | A    | 0    |
| B      | B    | 0    | C      | C    | 0    | D      | D    | 0    |
| C      | C    | 0    | G      | G    | 0    | B      | B    | 0    |
|        |      |      | E      | C    | 32   | E      | D    | 1    |

- If node D sends a message to node A, node C will attempt to forward the message to node A's MAC address even though A is no longer present on the V-LAN segment. However, the 802.11 MAC layer collision avoidance mechanism will not send the packet until it receives a CTS. After a set number of attempts (default is 7 attempts for RTS and 4 for DATA) the packet is dropped and the upper layers are notified. The node's VMTS agent removes this zero-hop neighbor from its table, which will prompt the route discovery process on the next attempt to send to the destination.
- Assuming no 802.11 CA mechanism and since traffic may be both UDP and TCP, it is possible that no transport layer acknowledgement from node A would be expected.
- There are at least two circumstances that will stop C from forwarding to A. Node A is discovered on another part of the network and node C learns of this discovery; or the lifetime of node A's entry in node C's table expires.

Part V: Consider what occurs if Node A reappears on node H's interface. Node H's current table entry looks like:

| NODE H |      |      |
|--------|------|------|
| DEST   | NEXT | HOPS |
| A      | G    | 2    |
| G      | G    | 0    |
| I      | I    | 0    |

- Node A moves while sending traffic addressed to previous neighbors but after a CTS has been received by the next hop.
- Assume node A doesn't realize it has moved out of node B and C's transmission range and attempts to send a message through node C for another node – say node E.
- Node H, however, is the only node that hears the message
- Node H will update its table with node A as a zero-hop neighbor and will check its table for node C to see if C is a zero or one hop neighbor. Since there is no entry for node B, node H will check for destination E. Since there is also no entry for destination E, node H will retransmit the message with node E's address as the destination and the MAC broadcast address as the next hop.
- Nodes G and I will retransmit, as will node J. Since node B has an entry for node E it will forward the message through node C without broadcasting. Each node along the path will increment the hop count and



update their table entry for A. The route through nodes H-I-J becomes the new shortest route.

- If E responds to the message, it will return through the shortest path with a valid lifetime. In this case, if the old path through D still has a valid lifetime E will use the old path since it is shorter. However, if E doesn't respond until after the second copy of the message arrives through node D then node E will have an updated longer route through node D thereby causing node E to send through node J.

## 6 Simulation environment

In an effort to better understand the ad hoc networking environment; to discover the strengths, weaknesses and performance characteristics of the various supporting protocols; and to investigate new concepts in ad hoc routing, we performed simulation analysis of existing ad hoc network protocols. We also attempted to model and simulate the proposed VMTS protocol. We chose the ns-2 simulator to conduct this analysis. The primary reasons for using ns-2 were its support of a multi-hop wireless environment and inclusion of four commonly referenced and thoroughly researched ad hoc protocols. Other advantages to the ns-2 simulation environment include its model of a physical layer, network interface and MAC layer that is suitable to represent our network template.

We used the ns-2.1b9a version of the simulator for the AODV and DSDV protocols but since the DSR protocol failed validation tests with this version, we switched to the older ns-2.1b7 version for the DSR simulations.

### 6.1 ns-2 simulator

The ns-2 simulator is a discrete event network simulator developed by the University of Berkeley and the VINT project and has been contributed to by many authors. Of significant note is the extensions made to ns by the Monarch Group from Carnegie Melon University. The CMU extensions, which have since been included in the base ns simulator, enabled a more accurate simulation environment for multi-hop wireless networks. CMU included a model for the physical layer, support for MAC protocols and added ARP functionality to address some of the challenges of operating in a wireless environment. [BRO][FAL]

#### 6.1.1 Physical layer

ns uses a physical layer model that models the effects of radio propagation on receive signal strength and capture. The signal propagation model uses a combination of free-space attenuation at near distances and two-ray ground attenuation at farther distances. Figure 6-1 shows the two models used. [FAL]

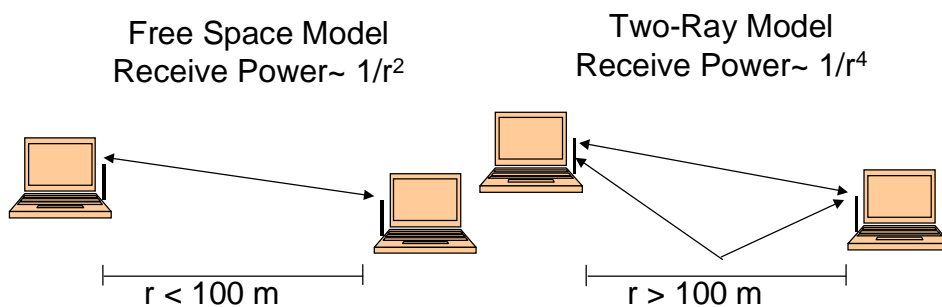


Figure 6-1 Radio propagation model

### 6.1.2 Network interface

The ns network interface simulates a hardware interface that is similar to that of the direct-sequence spread spectrum (DSSS) Lucent WaveLan radio. The interface is used to access the channel and is subject to the propagation effects discussed in the physical layer. The interface attaches to each packet additional data that is used by the simulator to calculate the received signal strength and to determine if a packet should be captured by the receiving interface. The data stamped on the packet includes power levels, wavelength, antenna gain etc. The receiving node network interface calculates the receive power based on this data and the distance between the nodes. This receive power is compared with the receive power of packets received simultaneously or packets that were already received by the interface but not completed. A comparison is made between the power levels to determine if a collision has occurred or if capture of one of the packets is possible. [FAL]

### 6.1.3 MAC layer

The ns MAC layer implements the 802.11 distributed coordination function (DCF). This provides carrier sense multiple access with collision avoidance (CSMA-CA) through a RTS/CTS/DATA/ACK pattern for unicast packets. Broadcast packets use CSMA without the collision avoidance mechanism. We have discussed in detail this function and the effect of the ns MAC layer in other sections of this document. [FAL]

### 6.1.4 Link layer

The link layer is responsible for simulating data link protocols if applicable to the simulation environment. It also resolves IP addresses to MAC addresses through an ARP lookup procedure. If a MAC address is not known, the link layer implemented for the wireless environment buffers the packet and broadcasts an ARP query. There is a buffer for a single packet for each unknown destination. If subsequent packets for the destination arrive at the link layer the earlier packet

is dropped. Once a MAC address for the destination is known the ARP module inserts the address into the MAC header. [FAL]

## 6.1.5 Mobile node object

### 6.1.5.1 Types of ns MobileNode objects

At the core of the ns wireless simulation model is the MobileNode object. The MobileNode is like the basic nsNode object with added functionality that supports wireless simulation environments. There are two types of MobileNodes included in the ns simulator package. The primary node design is used for DSDV, AODV and TORA ad hoc protocols. The schematic for this node is shown at left in Figure 6-2. The ad hoc protocol DSR uses a slightly different mobile node object due to its promiscuous mode operation. You should recall that promiscuous mode operation enables a routing agent on a node to receive all packets heard on the physical channel interface whether they are addressed to the node or not. The purpose of this is to allow routing agents to learn routing information from the packets on the channel. The schematic for DSR's SRNode is shown at right in Figure 6-2. [FAL]

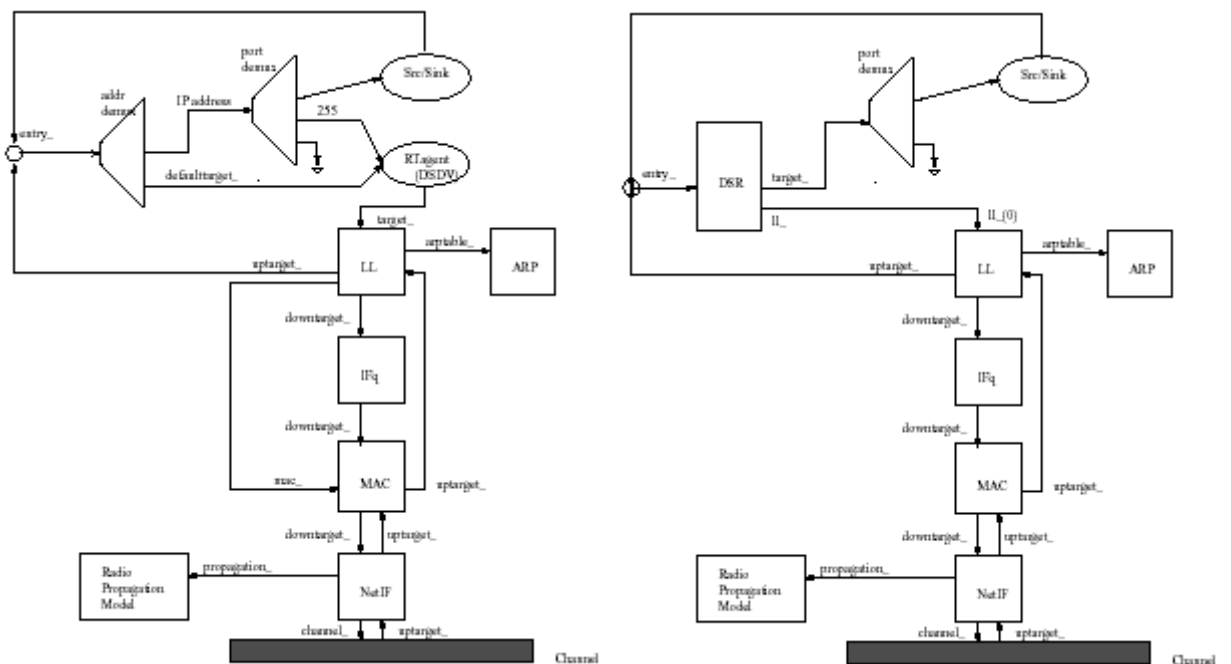


Figure 6-2 ns MobileNode objects (reprinted from *The ns Manual* [FAL])

Since VMTS performs routing at the MAC layer, we need to modify one of the existing MobileNode types. The VMTS routing agent needs to be lower in the stack, below the link layer and above the interface queue (IFq). We also need to suppress or fuse the ad hoc routing capability in the network layer to prevent

conflict with VMTS. We do not want to remove all reference to a network layer routing agent however, since VMTS would be part of protocol hierarchy and would still require a routing protocol similar to Mobile IP at the network layer. This upper layer protocol would accommodate the interface to structured networks and meet the challenge of MANET migration.

For simulation in ns, our protocol implementation reworked the base MobileNode object by adding a VMTS routing class that modifies the flow of the MobileNode and suppresses the network layer routing agent. This minimal adjustment to the base ns simulator accommodated VMTS but kept functionality of all other implemented ad hoc routing protocols. A final consideration for the modified MobileNode was the promiscuous mode capabilities that VMTS, like DSR, requires. However, since VMTS resides below the link layer, simply including VMTS as an uptarget from the channel allows it to receive all packets without incorporating SrNode modifications. The schematic of the VMTS MobileNode is shown in Figure 6-3.

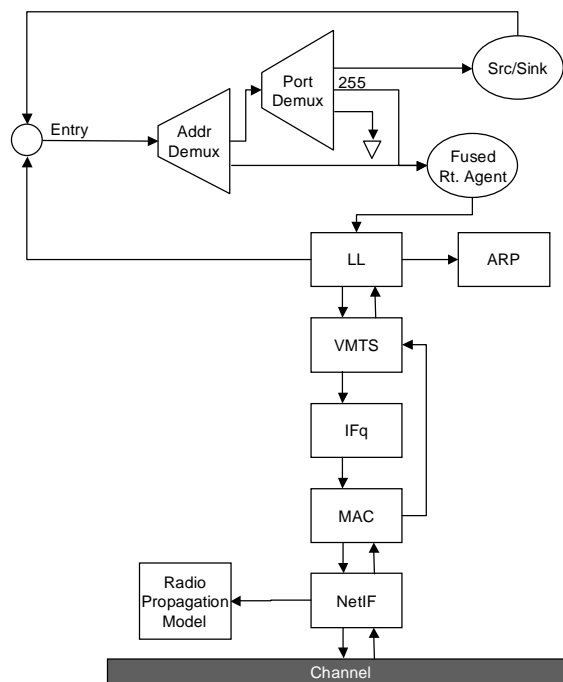


Figure 6-3 VMTS MobileNode object

### 6.1.5.2 VMTS MobileNode Operation

In the previous section, we described how we created a VMTS MobileNode object. We will now describe in more detail how packets flow through the object. Attached to each mobile node object are source agents that simulate applications and sink agents that receive packets. The source agent typically passes packets to routing agents that determine routes to other nodes in

the network. In our case, the MANET routing agent resides in the MAC layer so no routing is necessary at the network layer. Therefore, the fused routing agent in VMTS passes packets with their IP addresses from the application layer directly down to the link layer. The link layer uses the address resolution protocol to determine the hardware addresses that map to the message destination IP addresses.

In the basic nsNode object there is a direct resolution from IP to MAC addresses (IP=MAC). However, with the MobileNode objects, ARP will actually broadcast a request if the MAC address is not known. Since VMTS operates with MAC addresses, when the application layer hands down a packet, the link layer must invoke ARP prior to making the routing decision. However, if the channel hands up the packet to VMTS and is being forwarded then ARP need not be invoked.

Once a corresponding hardware address for the packet destination is found, the link layer hands the packet to VMTS. The VMTS routing agent determines a routing path for the packet and stamps it with appropriate MAC header and MAC\_Tag information. The packet is then sent to the interface queue, and stays there until a signal from MAC is received declaring the channel available.

The packet is copied to all interfaces at the time at which the first bit of the packet would begin arriving at the interface in a real physical system. Each network interface stamps the packet with its own properties, and invokes the propagation model. We should note here that the propagation model is invoked at the receiving end. The propagation model uses transmit and receive stamps to determine at what power the interface will receive the packet. The receiving network interface is left to decide whether the packet is received successfully or not. If successful, the packet is passed to MAC layer. If the MAC layer receives this packet as error-free and collision-free, it passes the packet to VMTS.

In the basic MobileNode, the packet would have entered a demultiplexer that determines whether the packet should be forwarded again or if it has reached its destination node. Since VMTS operates below this function, it determines if the packet has reached its destination node. If so, the packet is sent to the address and port demultiplexers, which decides to which application to deliver the packet. If the packet needs to be forwarded, VMTS determines the next hop and passes the packet back down toward the network interface. This operation is repeated until the packet reaches its destination. [CEL][FAL]

## **6.2 VMTS Implementation decisions**

### **6.2.1 802.11 MAC distributed coordination function**

As we have described earlier, CMU's Monarch group implemented in ns-2 the 802.11 MAC layer DCF, which provides a medium access protocol necessary for a multi hop wireless environment. The ns class CsmaCaMac extends the MAC layer for the wireless environment by adding a collision avoidance

procedure. The procedure sends a RTS/CTS/DATA/ACK pattern for all unicast packets and a DATA pattern for broadcast packets. [FAL]

This capability is relatively effective in avoiding collisions due to the hidden terminal problem common to wireless environments. The mechanism however, undermines a portion of the operation of the VMTS protocol. It was our desire to expand VMTS to perform the complete set of MAC layer functions in addition to routing. However, due to time constraints we were unable to develop a collision avoidance mechanism to complement VMTS. Therefore, in using the 802.11 DCF that is already implemented we discovered that there is overlap in functionality. In the original VMTS protocol design it was not necessary to use the MAC broadcast address (FF:FF:FF:FF:FF:FF) for route discovery messages since VMTS uses mechanisms that are broadcast in nature. VMTS nodes capture all packets on the interface and use the request bit in the MAC\_Tag to determine the necessary actions at each node.

The ns 802.11 DCF implementation, on the other hand, expects that all unicast packets (packets not using the MAC broadcast address in the MAC header destination address) to be intended for an immediate neighbor. The MAC layer DCF protocol sends a RTS packet addressed to the destination listed in the MAC header destination address. The MAC layer expects that the destination will hear the RTS and will respond with a CTS. If implemented in its original form, the VMTS route discovery messages would be considered a unicast packet, never receive a CTS and subsequently would never be sent.

Using the MAC broadcast address for route discovery messages was not difficult requiring only slight adjustments to the protocol. It does however limit the effectiveness of the Re-associate procedure and necessitated additional handlers for ARP packets. The re-associate procedure will only be invoked if a node receives a CTS from the destination but moves out of range of the destination before it sends the data.

### **6.2.2 ARP and source broadcast messages**

Now that discovery messages use the MAC broadcast address we needed to discern source broadcast messages from discovery messages. To do this we added a source broadcast bit to the MAC\_Tag. This indicator ensures VMTS routing agents at each node will send upper layer initiated broadcast messages up to higher layers while still propagating the messages to other nodes as is required.

### **6.2.3 Adjustable parameters**

VMTS has three parameters that are adjusted to improve the overall performance of the protocol. Currently these parameters are constants set prior to simulation. An optimization to VMTS would be to dynamically adjust them based on topology, mobility, or traffic conditions. The parameters are; the table (route cache) entry lifetimes `New_life` and `Short_life`; and the timer value for the

full path discovery mechanism. The default value from which we began simulation and the rationale for them is discussed in the remainder of the section.

### 6.2.3.1 New\_life parameter

The New\_life is value entered into the LIFE field of the routing table when caching a new destination entry. The value is adjustable and must be controlled as one of the simulation parameters.

- Rationale: We want to maintain a timer for each cached destination entry to identify the entry's age. This value facilitates selection of the most appropriate route (Next\_hop) and removal of stale entries. The value must be of sufficient duration to prevent excessive route discoveries but short enough to minimize caching of stale or broken routes.
- Estimated appropriate value: **100s**.
- Precedence: In simulation, the mobility patterns will dictate how long a cached route remains valid. The range of possible values extends from at most the entire duration of the simulation down to almost 0s. However, prior simulation studies have used values of 300s for the length of time that an AODV route is considered active and 15s for periodic table updates in DSDV. Another approach to consider is to examine the range of the radio model and speed of the nodes. Since the radio model in the ns simulation uses a range of 250m one would expect that these nodes are not modeling vehicle-mounted radios, which normally have a range of 15-25 km, but are modeling man portable radios. Therefore, an appropriate node speed to consider is 1m/s or 3.6km/hr. Eliminating the case of parallel node movement and assuming constant mobility, the best-case connection duration occurs when two nodes approach each other along the same vector. The nodes remain "within range" for the time it takes to traverse 500m. The worst case occurs when node vectors present minimal overlap of radio range fans. Considering a node speed of 1m/s, the cache entry lifetime falls in the range of 0 to 500s. If we were to increase node speed to 20 m/s (72km/hr), as in many studies, the cache entry lifetime falls to a range of 0 to 25s.

### 6.2.3.2 Short\_life parameter

The Short\_life value is entered into the LIFE field of the routing table when adding a temporary destination entry during a route discovery procedure. The value is adjustable and must be controlled as one of the simulation parameters.

- Rationale: We want to maintain a timer for the temporary destination entry that prevents recursive broadcasting of the same discovery message. This entry stops additional broadcasts to a destination currently in a discovery mode for a period equal to Short\_life. The value must be of sufficient duration to prevent a node from sending a discovery message



that reappears on its interface but short enough as to not cause undue delay to new packets for the destination.

- Estimated appropriate value: **15ms**.
- Precedence: In simulation, the network topology and the potential for loops dictate the appropriate length for this value. The appropriate value must allow the discovery message to propagate a sufficient number of hops to minimize the chance of reappearing on the nodes interface. Values to consider include end-to-end delay and per hop delay. Prior simulation studies have used constants of 30ms for the DSR timeout for a non propagating search which is equal to the time to send a request and receive an ACK for an immediate neighbor; and 150-250ms HELLO and ACK aggregate delay using TORA. Simulation results have also shown an average end-to-end packet delivery delay of 10ms to 1.4s.

### 6.2.3.3 FPD\_value parameter

The FPD\_value is used with the full path discovery optimization in the routing agent process. Recall that an FPD packet adds the addresses of all nodes through which it passes to its MAC\_Tag. The value is adjustable and must be controlled as one of the simulation parameters.

- Rationale: We want to set a limit equal to FPD\_value for the FPD optimization timer (FPD\_timer). Once FPD\_timer reaches this limit, the next message sent by the node's routing agent would enact the full path discovery keeping the addresses of every node through which it passes. The value must be of sufficient duration to prevent excessive overhead caused by the increased header size and short enough to cache and update table entries.
- Estimated appropriate value: **15s**.
- Precedence: Prior simulation studies have used 15s for periodic table updates in DSDV.

## 6.3 Description of ad hoc protocols implemented in ns

There are four ad hoc network protocols currently integrated into the ns software. There are also several more implemented as contributed code but are not integrated in ns. In this section, we describe the characteristics and behavior of three protocols integrated in ns that are quite different but have particular functions that are common to VMTS.

### 6.3.1 DSDV

Destination sequenced distance vector (DSDV) is a Hop by hop distance vector protocol in which nodes make decisions on the optimal path with only the "distance" to the destination. Distance is typically measured in hops but it could be delay or other cost metric. DSDV nodes broadcast periodic routing updates to

maintain the routing tables. Each node's routing table lists the next hop for every destination. The "route" in the table is tagged with a sequence number (SN) that is used to ensure the freshness of the route data. Both the distance metric and the SN are used to determine the best route to use. A route to the destination with a higher SN is better but if the SNs are equal then route with lower metric is used. Each node advertises (broadcasts) an increasing even numbered SN for itself. For example, say node 'B' decides that the route to destination 'D' is broken, node B increases the SN for that route by one (SN now odd) and advertises the route with an infinite metric. Any node 'A' that routes through B adds the infinite metric to their route table. Node 'A' keeps this metric until it hears a new route to D with a higher SN. DSDV sends triggered updates with each new sequence number and new metric. [BRO]

### **6.3.2 DSR**

Dynamic Source Routing (DSR) uses source routing techniques that require Route Request (RREQ), Route Reply (RREP), Route Error (RERR) messages. When a node needs a route to a destination, it broadcasts a RREQ message first to its immediate neighbors and if that does not return a route, it floods the RREQ throughout the network. The destination or any node that knows the route to the destination returns a RREP to the source.

DSR uses route maintenance to detect topology changes that cause a break in a source route. When a node encounters a fatal transmission error, it replies to the source with a RERR message. The source may choose another route in its cache or initiate another RREQ.

No tables are necessary in intermediate nodes, however every node maintains a cache of source routes. No periodic route advertisements are necessary and no active neighbor detection mechanisms are used. DSR uses promiscuous mode capabilities of the physical layer to learn new source routes to destinations, repair existing routes, and provides opportunity to define more efficient routes. [BRO]

### **6.3.3 AODV**

The Ad hoc On-Demand Distance Vector (AODV)[27] routing protocol is basically a combination of DSDV and DSR. It uses on demand route discovery and route maintenance found in the DSR protocol and the hop-by-hop routing, sequence numbers and hello beacons of DSDV. It is purported to offer quick adaptation to dynamic link conditions, low overhead and efficient network utilization. AODV uses destination sequence numbers to ensure loop freedom at all times, avoiding problems (such as 'counting to infinity') associated with classical distance vector protocols. It builds routes between routes only as desired by source nodes and it maintains these routes as long as they are needed by the sources.

AODV builds routes using a route request/route reply query cycle. When a source node desires a route to a destination for which it does not have a route, it broadcasts a route request (RREQ) packet across the network. Nodes receiving this packet update their information for the source node and set up backward pointers to the source node in the route tables. RREQ contains the source node's IP address, current sequence number, broadcast ID and the most recent sequence number for the destination of which the source node is aware. A node receiving the RREQ may send a route reply (RREP) either if it is the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. Otherwise, it rebroadcasts the RREQ. Forwarding nodes keep track of the RREQ's source information to ensure subsequent copies of a RREQ received are discarded.

As the RREP propagates back to the source, each node along the path sets up the forward path to the destination. Upon receipt of the RREP, the source may begin to forward data packets to the destination.

AODV maintains a table with the destination IP address, destination sequence number, hop-count, next hop and a lifetime, which is the duration for which this route is considered valid. AODV also maintains an active neighbor list, which include neighbors that use this route entry.

AODV broadcasts periodic HELLO messages to perform route maintenance. Failure to receive three consecutive HELLO messages determines the failure of a link. Whenever this is detected, upstream nodes are notified with an UNSOLICITED ROUTE REPLY message, which includes infinite metric for the destination. The source must then initiate a new route discovery. A route is considered active as long as there are data packets periodically traveling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. AODV allows mobile nodes to respond to link breakages and changes in network topology. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination. After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery.

## 7 Simulation Methodology

Now that we have introduced the ns simulation environment and existing protocols, in this section we will portray our simulation methodology. In an effort to stay true to our operational goals while at the same time, conduct comparisons with published performance evaluations we chose as our primary network model a 50-node ad hoc network. Although, we present test results with smaller node configurations the predominance of the simulation results involve 50 nodes. For clarity, plots and analysis will indicate the number of nodes used. Also, all simulation results presented are from trials with 1Mbps link rates.

The remainder of the section will detail the topography, communication model, traffic model and metric used. We will also discuss coding issues, code verification and model validation.

### 7.1 Simulation topography

The network simulation was based in a 1500 x 300 meter flat grid topography. We chose a rectangular topography for two reasons. Although a square topography may allow more freedom of movement [JOH], a rectangular topography as used in [BRO] results in a larger average hop count between source and sink pairs as discussed in [CAM]. Since an evaluation of ad hoc protocols must truly test the effectiveness of the protocols forwarding ability and the subsequent impact of multiple hops on performance this seemed to be the appropriate choice. In addition, since the simulator topography is flat, by forcing longer routes we may achieve some of the effect that physical obstructions would place on a network. Secondly, in the military scenarios that motivated this research, actual deployments of troops and systems are rarely in a regular shaped area of operation and even less frequently in an area that is relatively square.

### 7.2 Movement model

In our simulation, fifty nodes are randomly placed in the topography and move in a pseudo-random manner by picking a destination point, moving to the destination at a variable but bounded speed and pausing at the destination for a variable but bounded period. We chose to simulate two average node speeds, 1 m/s with a .5 m/s variation and 10 m/s with a 5 m/s variation. We also simulated with five different average pause times, 10s, 50s, 120s, 500s, and 900s. The pause times simulated varied by up to 10% around the mean. Each movement scenario lasted for a period of 900s. Since the protocol performance may be sensitive to movement patterns [BRO], we generated 10 scenarios for each pause time and movement speed for a total of 100 movement scenarios. We used a movement generation program from [CAM] called *mobgen* versus the *setdest* program developed by the CMU Monarch group [BRO]. *Setdest* starts with all nodes stationary for the pause-time period while in *mobgen* scenarios the

nodes randomly select whether they begin as stationary or mobile. *Mobgen* also provides greater flexibility in varying pause time and speed during simulation runs.

### 7.3 Communication/traffic model

In an effort to test the scalability of the protocols, we chose to simulate using 10, 30 and 50 sources, each transmitting to an associated peer. To exercise as much of the network as possible, sources and sinks are assigned in a manner that minimizes the number of nodes that perform both source and sink roles. For example, in the 10 source simulations no node performs both roles. In the 30 source simulations, only 10 nodes have both source and sink attached.

Most papers that conduct performance analysis of ad hoc network protocols use constant bit rate (CBR) sources. CBR sources are sources that are constructed with a connectionless protocol (UDP in our tests) or other protocol that does not apply congestion avoidance or other link or destination feedback mechanisms. In [BRO], the authors explain that since TCP sources offer a conforming load to the network, the time at which packets are sent and the position of the nodes when packets are sent would differ. As this would prevent a direct comparison between protocols, they chose to use CBR sources. Since our goal is to measure how well a protocol would perform with *realistic* traffic conditions and evaluate the effective throughput of the network, we chose TCP sources and an FTP application.

In selecting packet sizes, we considered the reasoning in [BRO] to use 64 byte packets. They observed that congestion became a problem for all protocols tested when 1024 byte packets were used. This resulted in some nodes dropping all of the packets received for forwarding. **The cause of congestion stated was the lack of spatial diversity.** We believe that by nature, nodes in ad hoc networks tend to cluster and that lack of spatial diversity is an inherent characteristic with which protocols must contend. We will learn later in section seven that spatial diversity is not the only cause of these packet drops.

Since our focus is not on comparison of the statistics of routing protocol generated packets alone, but on discovering performance characteristics of ad hoc networks and the various supporting protocols in realistic conditions, we considered larger packet sizes. To develop our own understanding of the effect of packet sizes on network throughput, we ran tests experimenting with different sized packets. We started with a baseline scenario using both DSDV and AODV protocols. The scenario was a simple two-node network with one source in a 100x100 m topography. The results listed in Table 7-1 present throughput and utilization as a factor of packet size for each protocol. The throughput in Kbps is defined as the number of data bits received at the destination node application layer divided by the simulation time. Throughput calculations exclude transport, network and MAC layer overhead as well as duplicate TCP packets. Utilization is the throughput divided by the maximum link rate of 1 Mbps. The results in Table

7-1 show that in this scenario the throughput and utilization experienced marked improvement with larger packet sizes.

| Packet size (bytes) | DSDV              |             | AODV              |             |
|---------------------|-------------------|-------------|-------------------|-------------|
|                     | Throughput (Kbps) | Utilization | Throughput (Kbps) | Utilization |
| 64                  | 141               | 0.14        |                   |             |
| 256                 | 332               | 0.33        |                   |             |
| 512                 | 509               | 0.51        | 510               | 0.51        |
| 1024                | 671               | 0.67        | 676               | 0.68        |
| 1400                | 735               | 0.74        | 735               | 0.74        |

**Table 7-1 Effect of TCP packet size on throughput (2 node)**

We further tested with our 50-node network with 10 source-sink pairs at 1 m/s and 120s pause-time and again with 50 sources. Since there are multiple sources in this test, the throughput shown in Table 7-2 is the network throughput, which is the combined sum of the application data received at all 10 or 50 destinations divided by the simulation time. The utilization is the network throughput divided by the shared medium maximum rate of 1Mbps. We will discuss the rationale behind this method of calculating utilization in the next section but for now the shared medium maximum rate assumes that all nodes are in one collision domain and a utilization of 1.0 represents full use of the 1Mbps shared link.

The results for the multi-source tests, shown in Table 7-2, are similar to that of the two-node, single-source tests in that larger packets provided greater throughput. The fact that the network utilization exceeded 1.0 with 1400 byte packets is not an error but demonstrates that the network is not a true, shared medium, but is able to exploit spatial diversity. In fact, it is more surprising that **more** of the tests did not produce a greater than 1.0 utilization. Considering 50 source sink pairs, distributed among nodes with 1Mbps bi-directional links, in a 1500x300 m topography you should expect a much more effective use of network resources. As we will see throughout the paper, this is a significant discovery and an unfortunate reality for ad hoc networks in their current state.

Attempts with larger packet sizes such as 2800 bytes failed to show significant improvement in throughput and sometimes resulted in slightly lower performance. Segmentation did not occur at any of packet sizes tested and therefore was not a factor.

| 10 sources          | DSDV              |             | AODV              |             |
|---------------------|-------------------|-------------|-------------------|-------------|
| Packet size (bytes) | Throughput (Kbps) | Utilization | Throughput (Kbps) | Utilization |
| 512                 | 588               | 0.59        | 527               | 0.53        |
| 1400                | 694               | 0.69        | 757               | 0.76        |
| 50 sources          | DSDV              |             | AODV              |             |
| 512                 | 718               | 0.72        | 664               | 0.66        |
| 1400                | 1089              | 1.09        | 1062              | 1.06        |

**Table 7-2 Effect on TCP Packet size (50 node)**

It is clear that the overall utilization of the network resources has improved with the larger packet sizes but we needed to consider the potential penalty of congestion at some nodes. We examined the traces of AODV for the 50-source scenario used above and found that with 512-byte packets 16 sources failed to deliver more than 2% of the trial's mean source throughput. In addition, 19 sources failed to deliver more than 2% of the mean when a 1400 byte packet was used. This would seem to confirm the conclusions in [BRO] regarding congestion with large packet sizes. However, when we decreased the packet size to 64 bytes the results were equally poor with 20 sources failing to deliver more than 2% of the mean source throughput. To further analyze this we determined which nodes were not performing and learned that 16 of the poor performing nodes were consistent across tests of all three packet sizes. Nearly identical results were observed with DSDV. Since these tests showed no conclusive evidence that larger packet sizes increased congestion, we pursued our simulations with 1400 byte packet sizes in accordance with our goal of challenging the protocols with more realistic traffic.

#### **7.4 Metrics**

We observed that in many published papers regarding MANET protocol performance the metrics commonly used were packet delivery ratio, routing overhead in packets or bytes, and path optimality. These metrics were used in one of the first performance evaluations on ad hoc networks in [BRO]. The authors explained that the packet delivery ratio describes the loss rate seen by upper layer protocols and applications. This in turn affects the throughput that the network can achieve. Routing overhead is said to indicate the protocol's scalability and performance in low bandwidth environments, and path optimality is suggested to indicate a protocol's ability to efficiently use network resources. To determine the appropriate metrics for our goals we first considered what would be the operational requirement of the notional network. The network must **effectively and efficiently use the available resources, providing an established minimum throughput to all sources**. It should also provide these qualities in the most demanding of traffic and mobility conditions.

Given these goals we found the status quo metrics, albeit interesting, insufficient for a proper evaluation. To be effective, a network must deliver data from source to destination. The most effective way to measure this is to calculate the throughput. For consistency and clarity, we developed the following measurement protocol and related terms. We will use these terms throughout the remainder of the document. Because of learning that the throughput was going to be much lower than expected during our preliminary tests with packet sizes, we chose to evaluate the network as if the medium was shared by all nodes. Since each maximum link rate is 1Mbps, the *shared medium maximum* data rate is 1Mbps. The *network throughput* is the total connection rate or the sum of all bytes received by all sinks during the trial, divided by the simulation time. The network throughput includes the sum of all connection data transfers whether they are shared or not. In other words, it is possible to have a network throughput that is larger than the shared medium maximum rate if the network is able to take advantage of spatial diversity. The *average network throughput* is the arithmetic mean of the network throughput for the set of trials. In the last section, we referred to *network utilization*. This is equal to the average network throughput divided by the shared medium maximum rate. Again, because of spatial diversity it is possible to have network utilization greater than one.

Routing overhead is a relatively useful measurement of efficiency, but to evaluate overhead you must not only consider packet overhead but also byte overhead. The impact of 1,000 64-byte packets on a network's resources is arguably not as severe as 200 512-byte packets. Therefore, we captured results for overhead in terms of both packets and bytes.

In an attempt to evaluate fairness, we also computed the standard deviation of the bytes received at the sources over the set of trials. For continuity with previous papers and to examine routing accuracy we will report the packet delivery ratio as computed by the quotient of the number of data packets received and the number of data packets sent.

## 7.5 Coding

Most of the actual coding to implement the network model and to simulate the pre-existing ad hoc protocols was already accomplished and is included in the various ns builds. However, we created several interface scripts to control the simulation, iterate and automate the trial execution and to pass the necessary parameters to ns. We also created scripts to process the trace data during run time and to perform post processing of the trial output. Examples of the primary scripts are included in Appendix A to this document.

Coding of the VMTS protocol and modification of ns to accommodate MAC layer routing was tasked to a team of WPI graduate students as a class project for a course on High Performance Networks. Although the protocol model was able to route packets in a simple 2-node test, the group was unable to correct all segmentation faults at the time of this writing.



## **7.6 Verification and Validation**

### **7.6.1 Verification**

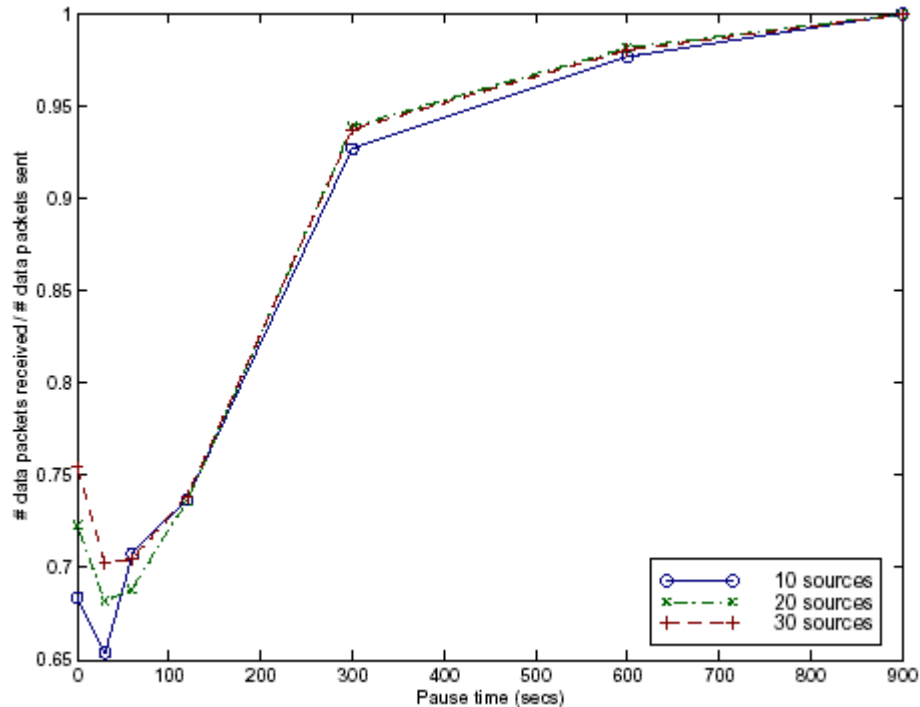
After the run scripts for the existing protocol evaluation were debugged, we proceeded to verify the proper operation of the code. We ran several tests on all protocols and examined the trace output. These tests ranged from the simple two-node network to the full 50-node network with varying number of sources. Since the protocols each contain several thousand lines of code on top of hundreds of thousands used in the related ns files we could not verify all aspects of the model. However, a few key aspects of the model were of concern. The first area of concern to examine was the MobileNode object and its handling of data packets. We examined the trace output, followed the life of various types of packets as they were generated and passed down through the protocol stack, and forwarded through the network. We examined ARP, data, routing and MAC packets in order to verify that proper handling and forwarding occurred. Another concern was node and source generation and nodal mobility. A useful tool to verify these aspects of the code is include with ns and is called the Network Animator (NAM). NAM provides a graphical interface demonstrating the mobility of nodes and traffic exchanges between them. Using this tool we were able to verify the generation of the nodes, initiation of movement and proper source sink interaction. Other concerns included the bandwidth and transmission range settings in ns. We have seen conflicting descriptions in publications and on user lists regarding which default values are set in the code. For example, the default bandwidth in ns version seven is 2Mbps and while the default for ns version nine is 1Mbps. Therefore, we compared tests with the default settings with tests in which we physically set these parameters. We were able to verify that the link bandwidth was set at 1Mbps while the defaults transmission range is set to 250m for all simulation runs.

### **7.6.2 Validation**

Since we are modeling a particular radio, the last two verification concerns can also be considered as validation concerns. Since we did not have access to a physical system we could not validate that ns correctly modeled a DSS WaveLan radio or any of the other physical characteristics of an ad hoc network. Fortunately, this validation has already occurred in [BRO] which also provides simulation for CBR sources in a 50-node ad hoc network. Therefore, we reproduced the tests conducted in [BRO] using our model and compared the results.

Figure 7-1 shows the packet delivery ratios (PDR) of the DSDV protocol with a 50-node network in a 1500 x 300m topology. Node speed was an average of 10m/s. The graph shows results from 10, 20, and 30 CBR source simulations.

You can see that the PDR ranges from 0.65 at high mobility to nearly 1.0 at no mobility (900s pause time).



**Figure 7-1 DSDV packet delivery ratios as published in [BRO]**

Figure 7-2 shows the results of validation trials on our 50-node network model using the DSDV protocol and similar movement scenario parameters. We ran 10 and 30 source simulation runs using CBR sources transmitting four 64-byte packets per second as used in [BRO]. Although the results are not an exact match, they are consistent in range and shape from the published results. We also ran these trials with the DSR and AODV protocols. Initially the DSR results were inconsistent with the results in [BRO] and other ad hoc network performance papers. The DSR PDR results ranged from 0.15 at higher mobility to 0.95 at low mobility. Other parameters examined were also inconsistent. Further examination of ns-user lists raised concerns that the DSR version in ns-9 had significant performance flaws. A suggested patch did not remedy the results. We then performed the trials using an older version of ns and achieved satisfactory results. The PDR results for DSDV, AODV and DSR are seen in Figure 7-3 and are consistent with the published results shown in Figure 7-4.

DSDV Packet delivery ratio (CBR, 10 m/s)

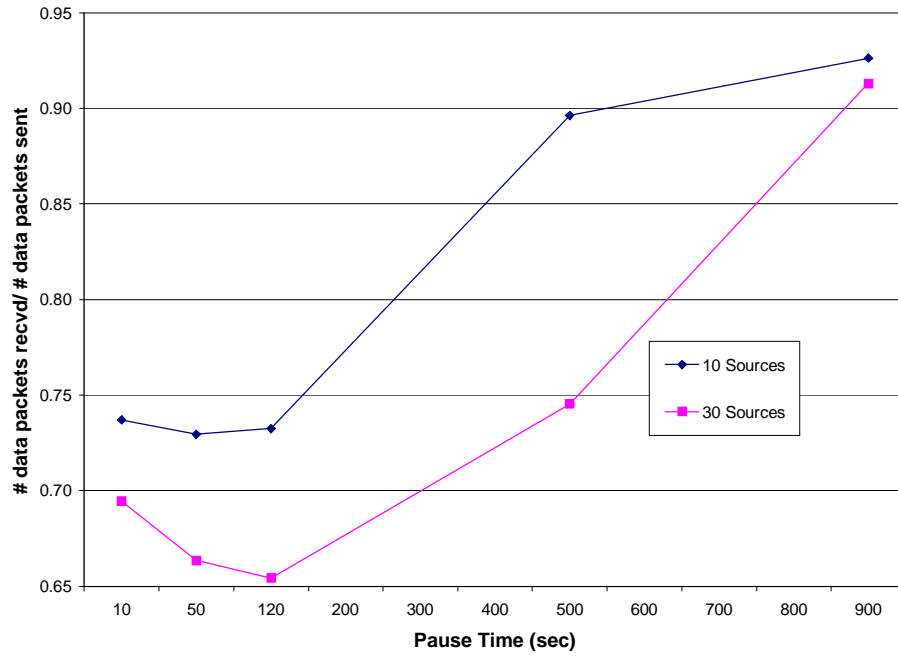


Figure 7-2 DSDV PDR in 50-node validation trials

Packet delivery ratio (10 src CBR, 10 m/s)

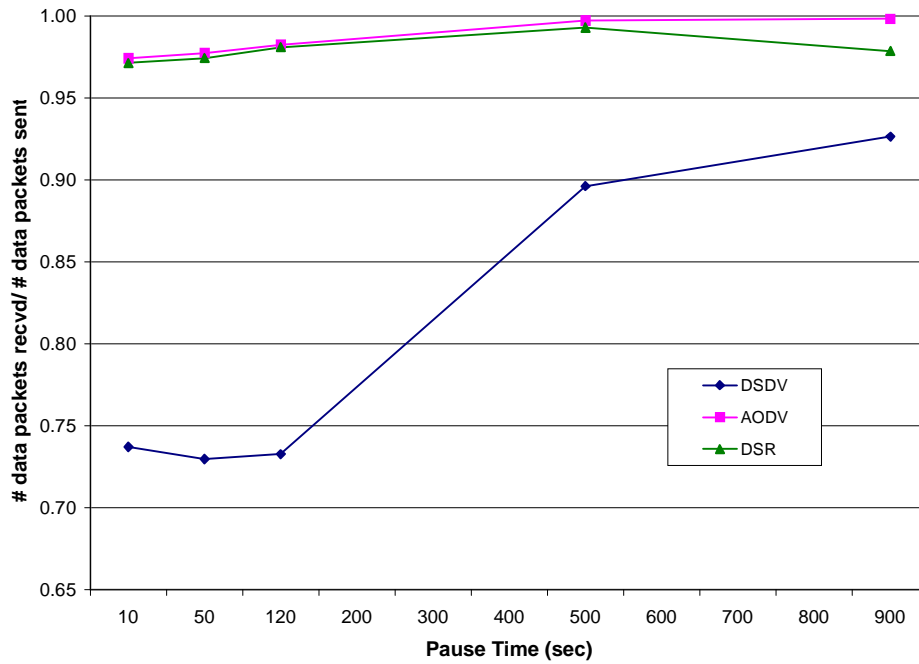


Figure 7-3 Protocol PDR comparison in 50-node, 10 source validation trials

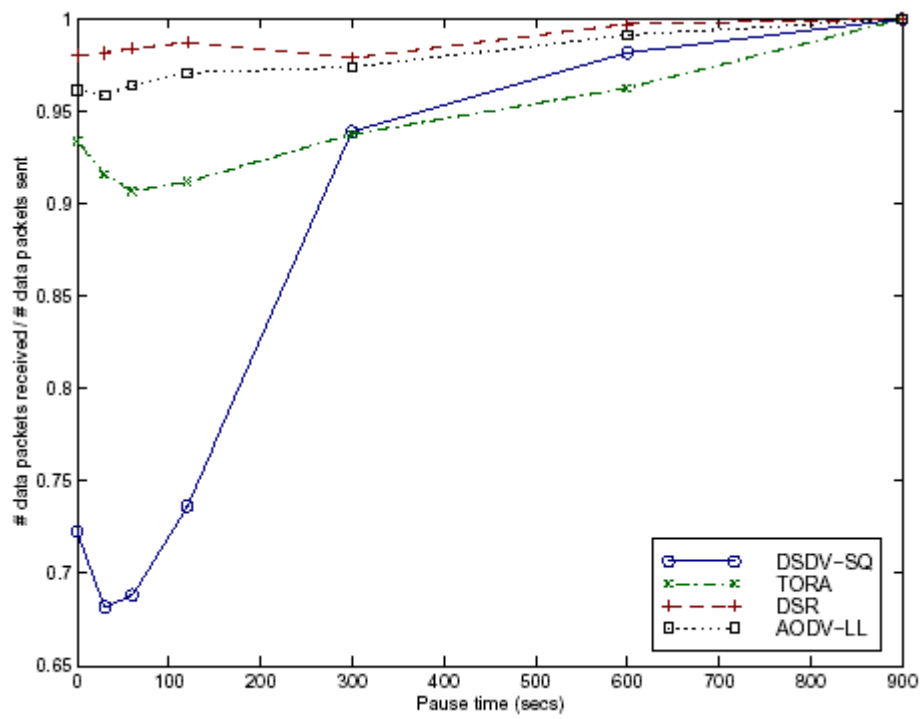


Figure 7-4 Protocol PDR values as published in [BRO]

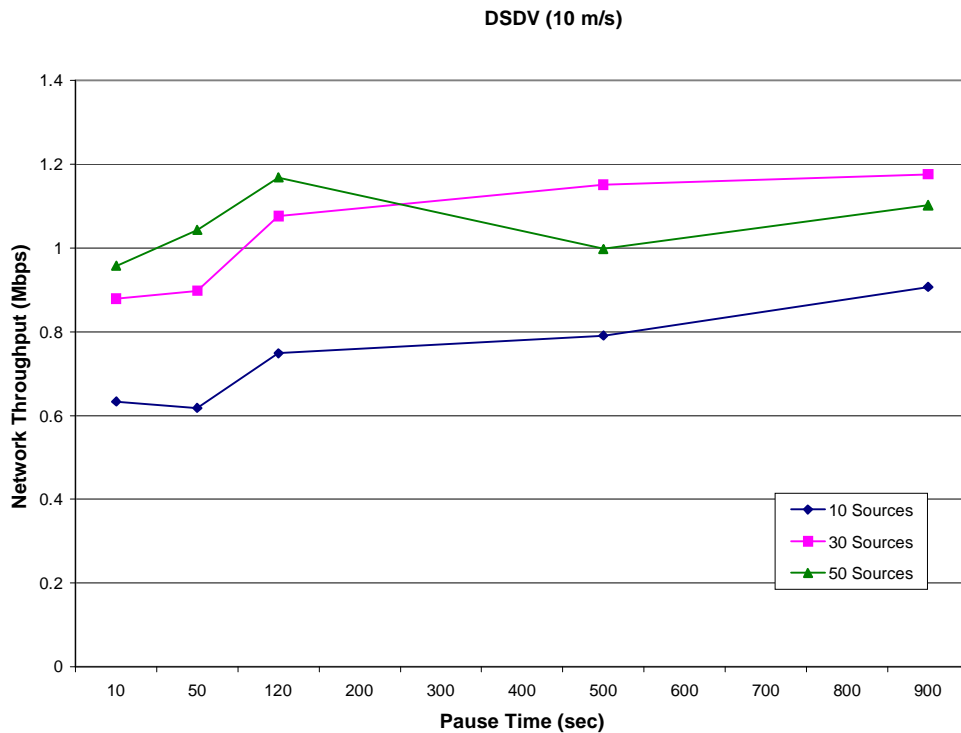
## 8 Routing protocol simulation results

In this section, we present a subset of the simulation results of the DSDV, AODV and DSR protocols in our 50-node network using 10, 30 or 50 TCP sources. We plot the performance metrics defined earlier using either single protocol comparisons of the three traffic models, or multi-protocol comparisons for a single traffic model.

### 8.1 Network throughput

As we discussed when defining the simulation metrics, the network throughput is the sum of all application bytes delivered to all of the sources during the simulation trial divided by the simulation time. **We should note that the maximum possible network throughput given best routing in a 50-node degenerate topology and 1Mbps links would be 10Mbps for 10 sources and 25Mbps for both 30 and 50 sources.** We can calculate these maximum throughputs simply by summing the links rate of each source-sink pair. Since each node in a 50-node, 10-source network only has a single source or sink, the links in a degenerate topology would be unidirectional and the source could exploit the entire 1Mbps link. It is easy to see that 25 sources in a 50-node network is the maximum number of sources for which all links are unidirectional. This gives a maximum network throughput equal to 25Mbps. Each additional source added must share a link with another source sink pair. Therefore, in a 30 and 50 source degenerate network the maximum network bandwidth is also 25Mbps.

In the following plots, we present the average network throughput for 10 different trials with the same parameters. Each of the ten trial groupings yields a single data point. The first plot in Figure 8-1 presents the average network throughput versus pause time from the simulation of the DSDV protocol at 10m/s average speed. Plotted values are for 10, 20 and 30 TCP sources. Throughput values range from just over 600Kbps for 10 sources at shorter pause times (high mobility) to nearly 1.2Mbps for 30 sources at 900s pause time. DSDV throughput results were unique among the protocols in that the 50 source trials performed worse than 30 source trials at low mobility.



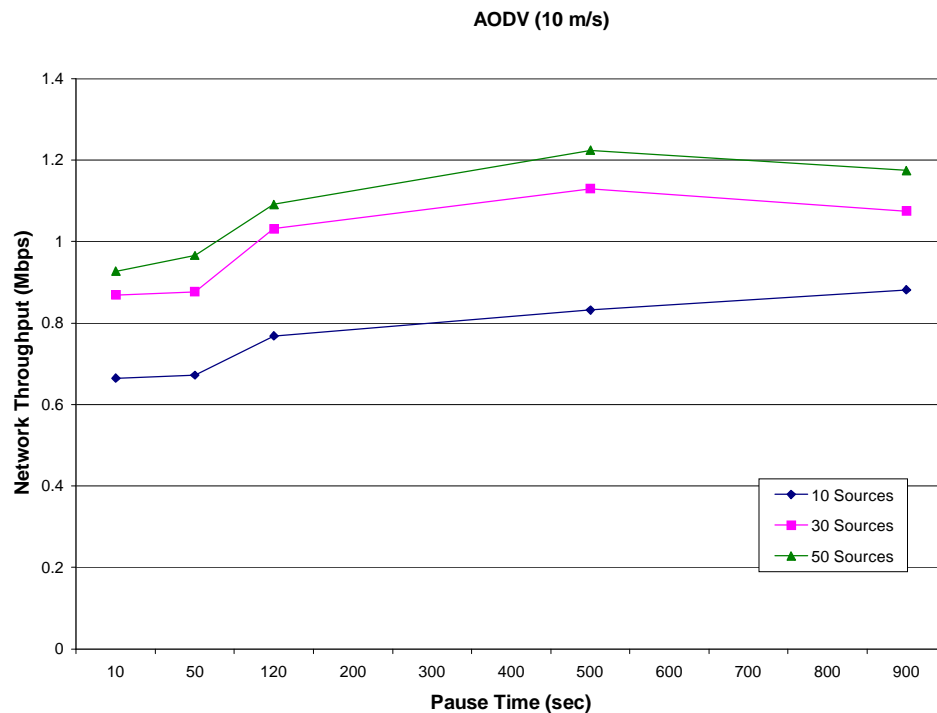
**Figure 8-1 DSDV avg. network throughput (50-node, 1Mbps links)**

Clearly, this network topology is not a degenerate network but these throughput results are startling nonetheless. There is obviously a certain achievable spatial diversity in a 1500x300m topography when each node has a transmitter range of 250m, however the network model failed to exploit it. We should add that additional tests performed on this network model with different transmission ranges failed to show a significant improvement in network throughput and the largest increases were at a cost to quality of service (QOS). We will discuss this QOS issue further in section 8.2 and again in section 9 but for clarity in understanding these results we will provide a brief explanation now. In the simulation trials using TCP traffic sources we witnessed that some sources failed to send any data to their respective sinks. As we decreased the transmission range, this issue worsened with fewer sources successfully sending data to their sinks. Table 8-1 shows results from these test trials using AODV in a 50 node, 50 active source network with 1Mbps links in which we adjusted the node transmission range from 70m to 350m. The results of the 250m-default range are in bold. An increase in range to 350m improved the number of successful connections but experiencing a slight decrease in throughput. Decreasing the range provided some improvement in network throughput, with the best results seen at a transmitter range of 150m. However, at this range only half of the active sources were able to successfully deliver data packets to their respective sinks.

|                                  |      |      |      |      |      |             |      |      |
|----------------------------------|------|------|------|------|------|-------------|------|------|
| <b>Transmission range (m)</b>    | 70   | 100  | 150  | 175  | 200  | <b>250</b>  | 300  | 350  |
| <b>Network throughput (Mbps)</b> | 1.39 | 1.83 | 2.17 | 1.57 | 1.45 | <b>1.23</b> | 1.23 | 1.11 |
| <b># Successful connections</b>  | 5    | 14   | 25   | 32   | 33   | <b>33</b>   | 33   | 39   |

**Table 8-1 Effects of transmission range on performance (50-node, 50-src)**

Continuing with the network throughput comparisons Figure 8-2 shows the results for the average network throughput using the AODV protocol. AODV performs better than DSDV with 50 sources at low mobility but worse with high mobility. AODV provided about equal network throughput with 10 sources and slightly worse with 30 sources. As with DSDV, performance generally decreased with greater mobility. Again, it is noteworthy that while AODV achieved between 650Kbps and 1.2Mbps, the maximum possible network throughput is 10Mbps and 25Mbps for 10 sources and 30/50 sources respectively.



**Figure 8-2 AODV avg. network throughput (50-node, 1Mbps links)**

DSR throughput results are plotted Figure 8-3. Again, performance of the network with 10 active sources was about equal to that of both DSDV and AODV. However, overall DSR performed better than both protocols with 30 and 50 sources. A comparison of the three protocols with fifty active sources is presented in Figure 8-4. It is clearer here that DSDV had the best performance at shorter pause times and DSR had the best at longer pause times.

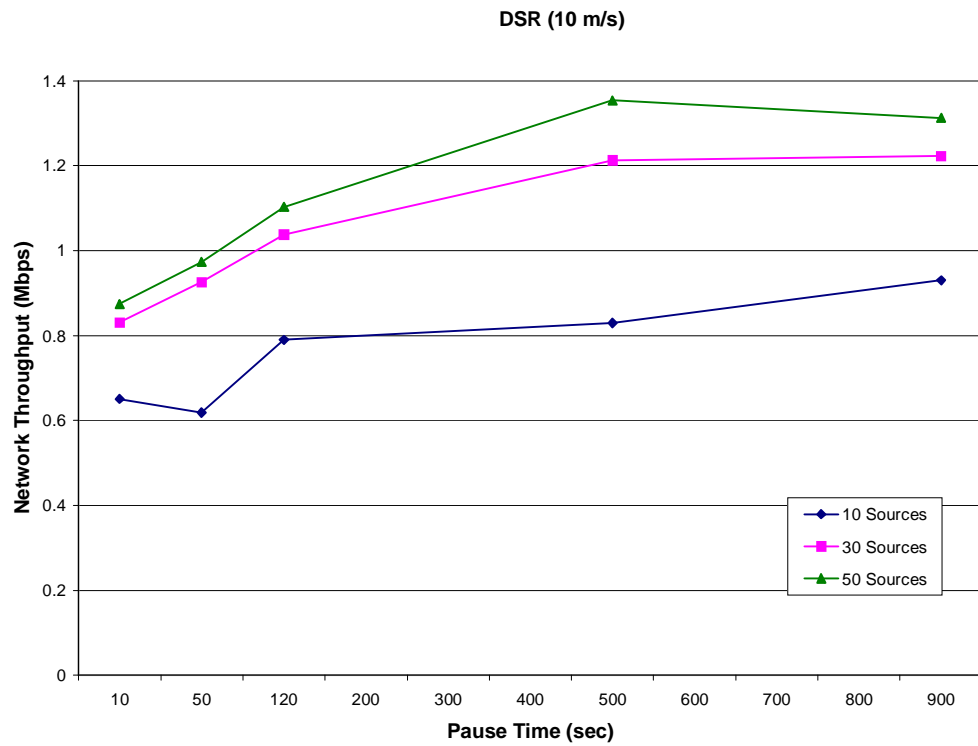


Figure 8-3 DSR avg. network throughput (50-node, 1Mbps links)

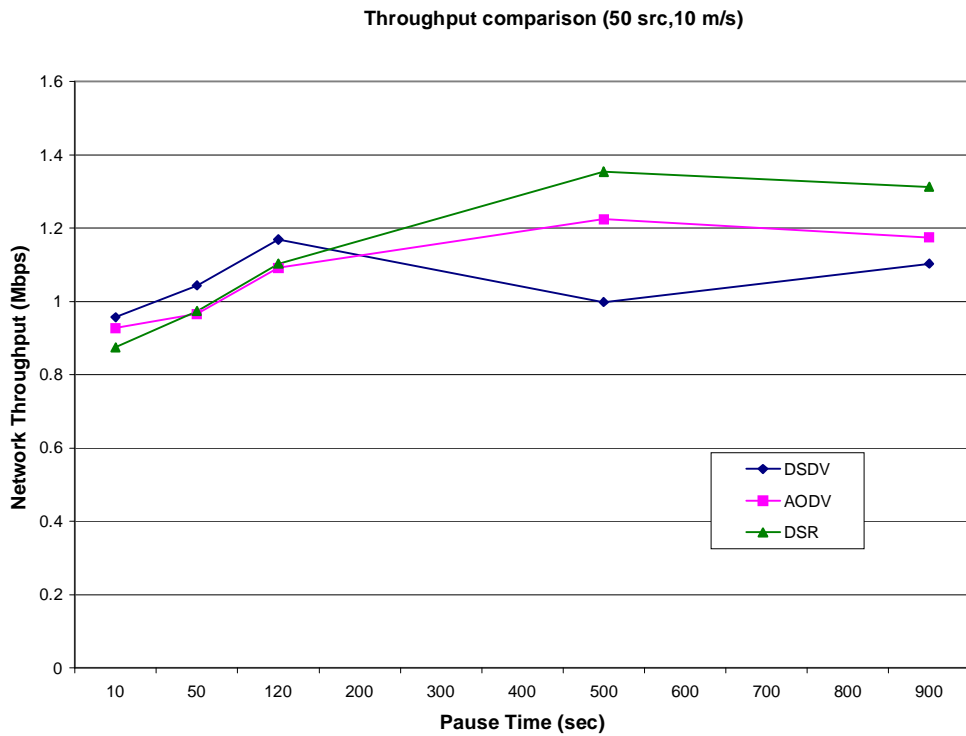


Figure 8-4 Comparison of 50-source throughput (50-node, 1Mbps links)



## 8.2 Standard deviation of source throughput

In an effort to measure the fairness of the network using the various protocols, we computed the standard deviation of the source throughput. The source throughput is the sum of the application bytes received at each of the destination nodes. The standard deviation computation is performed on all the source throughput values observed during the set of ten trials for each source, speed and pause time. To remove from our data the effects of the large throughput transient experienced by any TCP connection due to the slow-start algorithm, we stagger the start time of each source by 100ms and discard the first ten seconds of the simulation run.

Figure 8-5 shows the std. dev. in Kbps versus pause time for the three protocols with 50 sources at 10 m/s average speed. As shown in Figure 8-5, the standard deviation increases with the length of pause time. This is expected since the average network throughput also increased with the length of pause time. Since lower mobility allows some sources to achieve better throughput while other nodes achieve little, the range of possible source throughput values increase. Likewise, since the average network throughput observed by all three protocols is not very diverse we would expect similar values for the standard deviation. However as seen in the chart, the std. dev. of DSR source throughput is nearly twice that of AODV and DSDV. This alone does not suggest that DSR sources don't at least achieve an acceptable throughput but further examination of trace files showed that this is in fact the case. We examined the source throughput results of many simulation trials for all protocols and found that **several sources failed to deliver any data to their respective sink.**

One would expect that with so many non-productive sources the network throughput would decline. On the contrary, we found that up to a point, the more sources failing to deliver data the greater network throughput. There was a point at which such a high percentage of failed sources caused a decrease in the network throughput. However, this threshold was only when more than 50% of the active sources failed to deliver data. The rationale behind these results is that with fewer interfering sources a well functioning source-sink pair can come closer to exchanging their link bandwidth's worth of data. DSR was particularly vulnerable to this phenomenon and had a larger percentage of failed sources. This fact of higher throughput with more failed sources explains why DSR achieved better overall throughput results.

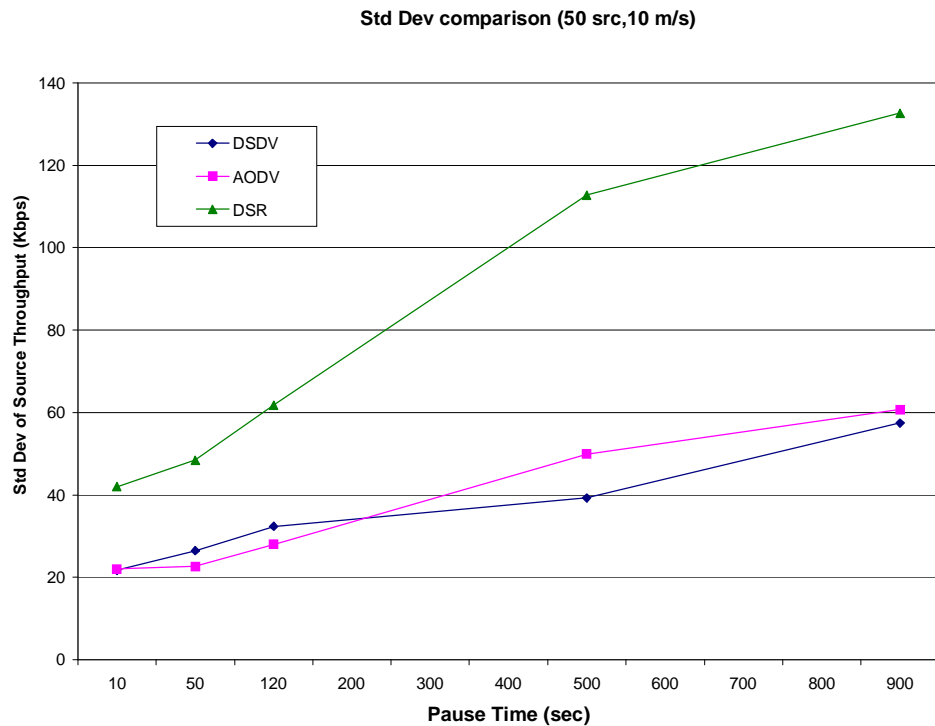


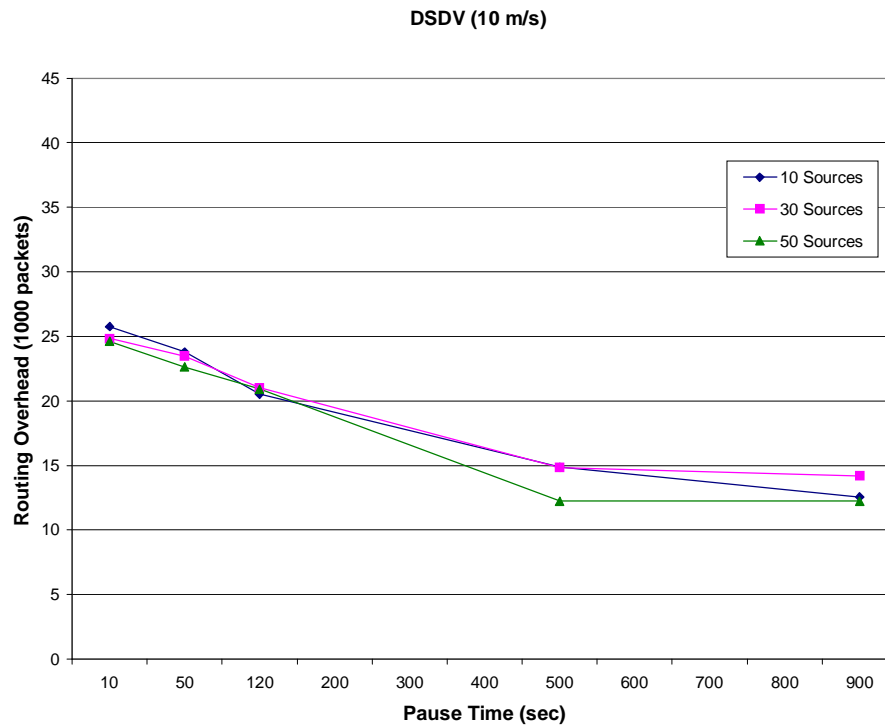
Figure 8-5 Comparison of source throughput std. dev. (50-node, 1Mbps links)

### 8.3 Routing Overhead

We examined the routing overhead incurred by each protocol during the trials. Many papers present only overhead in packets, but as discussed in the metrics section this only provides part of the information. The size of the routing packets is also very important as it has a direct impact on data throughput. However, many small packets incur a certain penalty in additional MAC headers and RTS/CTS exchanges of the MAC layer.

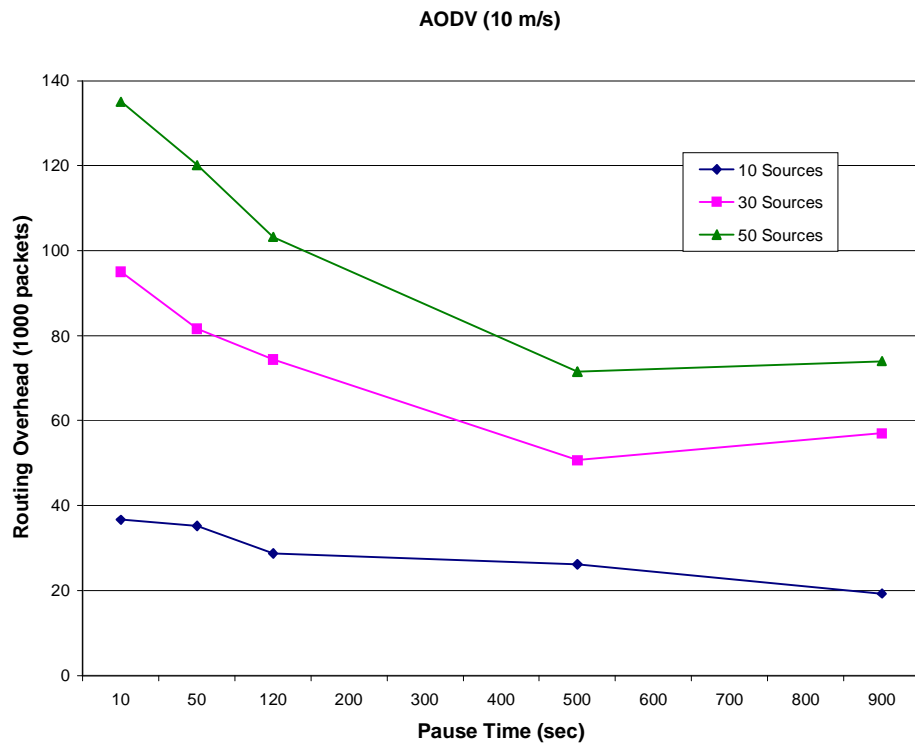
#### 8.3.1 Overhead in packets

During the simulation we counted the number of routing protocol initiated data packets at each node. To normalize the data we counted each instance that a routing packet is forwarded from one node to another as an individual packet. Otherwise, the penalty imposed on the network from a packet path of one hop would count equally as one with a path of several hops. Since each DSDV node broadcasts periodic updates, the number of routing packets is consistent between the various source densities. DSDV also use triggered updates, which occur more frequently with higher mobility. These results are clear in Figure 8-6.

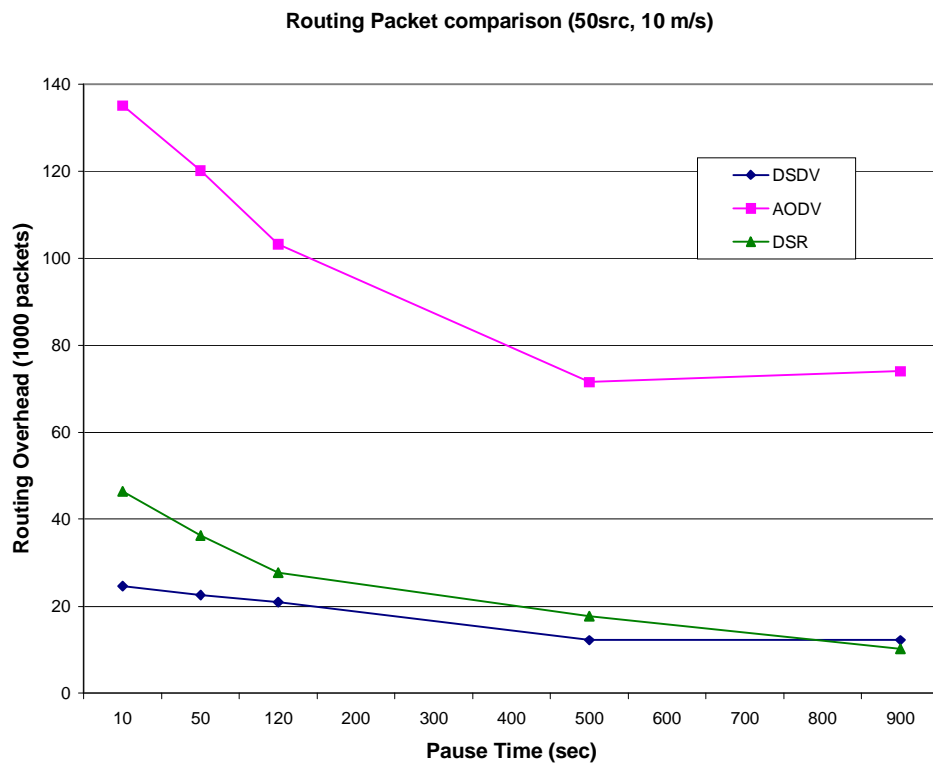


**Figure 8-6 DSDV routing overhead (packets) (50-node, 1Mbps links)**

AODV and DSR are on-demand protocols and exhibit characteristics of efficient routing at lower traffic and source densities and increased routing overhead with additional sources. Figure 8-7 shows a typical routing packet characteristic for on-demand protocols. DSR exhibits a very similar characteristic except that the magnitude is considerably smaller. Figure 8-8 presents the comparison of the three protocols. DSDV although periodic had the lowest number of routing packets at all pause times but 900s.



**Figure 8-7 AODV routing overhead (packets) (50-node, 1Mbps links)**



**Figure 8-8 Comparison of routing packet overhead (50-node, 1Mbps links)**

### 8.3.2 Overhead in bytes

Since routing packets are of various sizes, it is hard to see the impact on bandwidth utilization by routing packets from the previous results. Moreover, the results of routing byte overhead calculations shown in Figure 8-9 are very surprising. DSDV, which had the lowest number of routing packets during the trial, incurred the largest overhead in routing bytes with between 9Mbytes at high mobility and 3.5Mbytes at low mobility. DSR, which also had relatively low packet overhead, also has the lowest byte overhead.

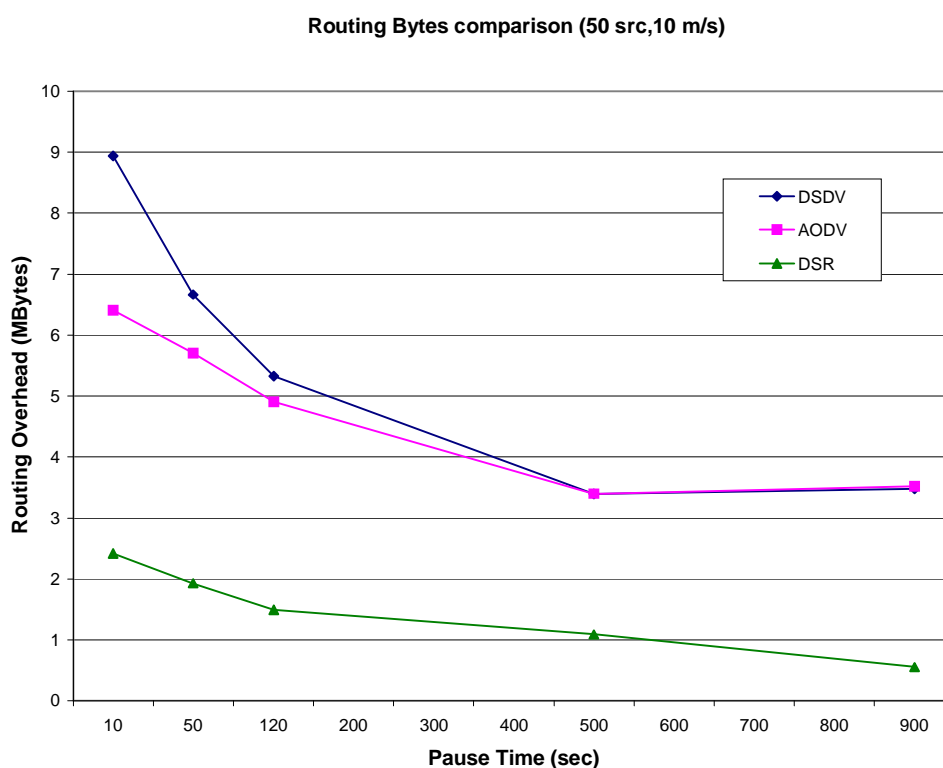
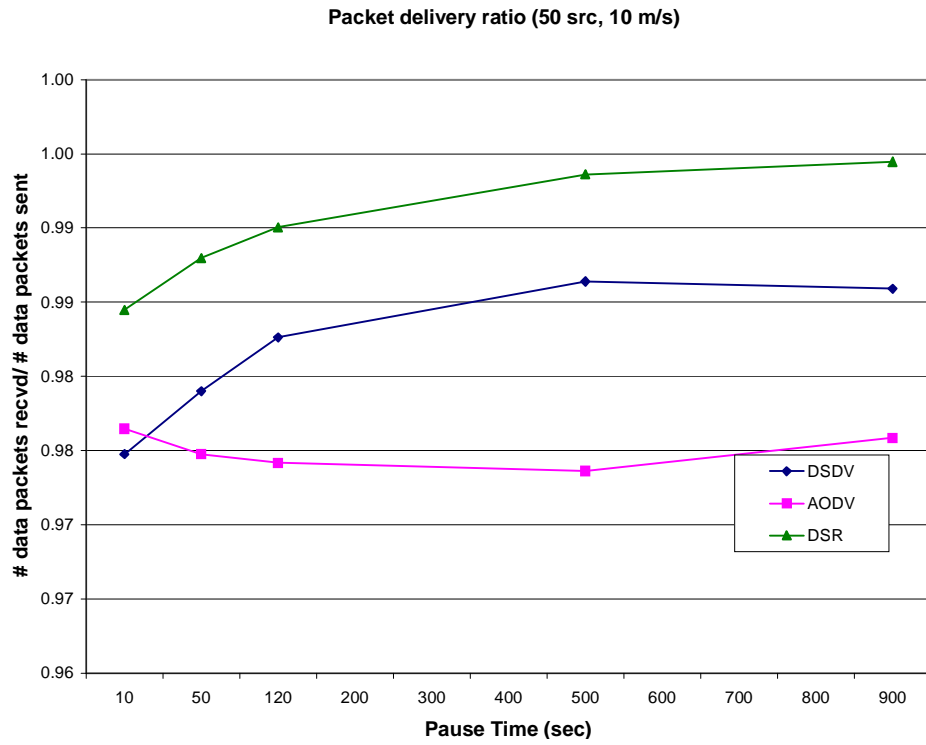


Figure 8-9 Comparison of routing byte overhead (50-node, 1Mbps links)

### 8.4 Packet delivery ratio

Packet delivery ratios are calculated by dividing the number of data packets received at the all of the node's routing agents, by the number of data packets sent by routing agents. Previously published papers using CBR traffic sources have presented the packet delivery ratio metric as their primary indication of routing protocol performance and accuracy. However, with TCP sources, packet delivery ratios become less of a factor in the protocol performance evaluation. Figure 8-10 shows that all protocols provide nearly 98% or greater PDR with TCP sources. Since the principal cause of dropped packets

is congestion, by using TCP sources with their congestion control mechanisms we greatly improve the chance that a packet will be successfully delivered. We will explore this further in the next section.



**Figure 8-10 Comparison of PDR with 50 TCP sources (50-node, 1Mbps links)**

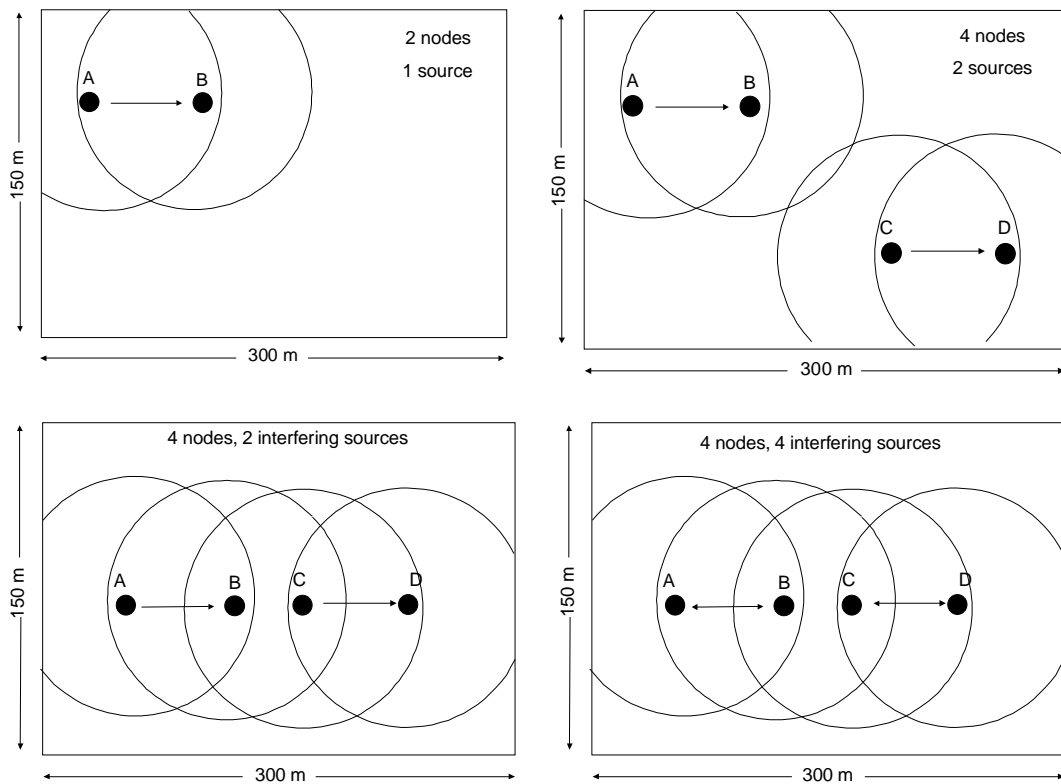
## 9 TCP performance and analysis

The purpose of the previous analysis was to model and simulate an ad hoc network, collect data, and compare performance metrics of various protocols, all in an effort to establish the characteristics of better performing protocols. However, the simulation results of the other protocols provided startling information regarding the performance of ad hoc networks. These observations required further investigation to reveal the causes. We were aware that we might discover that the causes of such poor performance were characteristics of the ad hoc network in general, or mechanisms of the routing protocols that are common to VMTS. In either case, these causes may, and as we will demonstrate, must affect the design of VMTS, else a VMTS equipped network will suffer the same bandwidth and QOS issues. It was this recognition that prompted us to suspend further implementation efforts of VMTS in exchange for investigation and attribution of causes.

An additional impact of our observations is the realization that previous ad hoc network studies were blind to important breakdowns in performance when faced with realistic traffic models. Hence, our purpose in this section, and in a larger respect the furtherance of the research in this thesis, is to conduct additional proofing tests that fix the cause of the throughput and QOS issues and to make these causes known.

### 9.1 Network Throughput observations

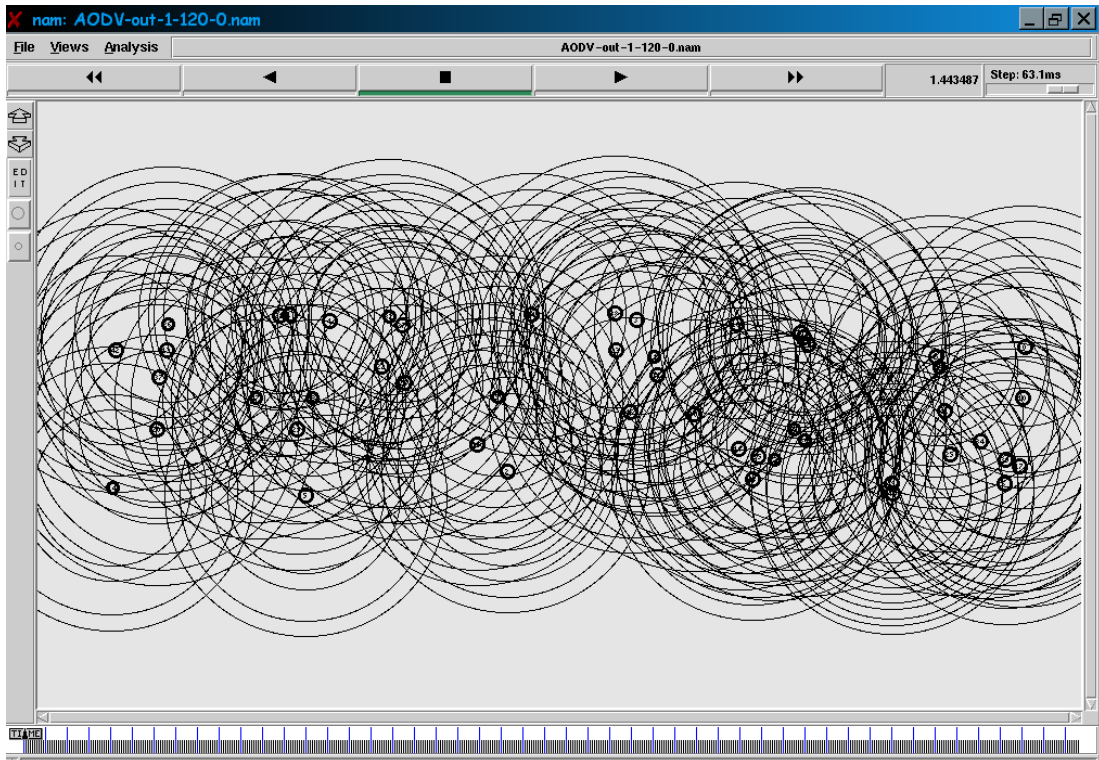
Although we would expect to be able to take advantage of spatial diversity in ad hoc networks, this was not realized in simulations involving the three protocols. We have seen from the throughput results in section eight that the total connection rate (network throughput) barely exceeds the shared medium maximum rate of 1Mbps in all protocols, even at low mobility. It is easy to demonstrate that interference is the underlying factor in the performance of the network and subsequently the operation of the protocols. Given the interference range and node distribution, the network was essentially reduced to a common channel. To further explore this assumption we performed four basic tests with two and four nodes. The test setup is seen in Figure 9-1. We first tested two nodes with one source and achieved a network throughput of 734Kbps. We then tested four nodes with two non-interfering sources and achieved a throughput of 1.47 Mbps demonstrating the benefit of spatial diversity. Four- node tests with two and four interfering sources produced throughputs of 736 Kbps and 723 Kbps respectively. These last two tests proved the effect of interference on throughput.



**Figure 9-1 Interference tests**

This problem is further aggravated by the use of broadcast mechanisms. Broadcast packets do not use RTS/CTS but a DATA/ACK pattern. When a broadcast message is sent the 802.11 MAC propagates the message throughout network without executing a RTS/CTS. An example of this phenomenon is seen in Figure 9-2. This figure was captured from an AODV simulation of the 50-node network. Since messages can only be captured if the transmission receive power is above the receive threshold the transmission of a broadcast packet can cause all transmissions in progress to fail.





**Figure 9-2 Effects of broadcast message on channel**

Another observed characteristic of the ad hoc network is what we have called *center of mass*. Since sources and sinks are randomly distributed throughout the topography the center of the topography becomes the network “center of mass”. Since all routing protocols use a shortest route mechanism and do not consider congestion when selecting a path, the center of mass also becomes the *center of routing*. The nodes located at the center of routing are more likely to become congested particularly if they too have active sources.

This behavior is more difficult to isolate and substantiate because simple packet counting techniques aren’t effective. Once a node goes into congestion the number of packets forwarded or sent from its source will reduce, thus skewing the results. Tests that combine the counting of forwarded packets, dropped packets and measuring delay through a particular node would provide interesting results. Applied graph theory would also provide a statistical analysis. However, by simply examining traffic flows in NAM, visible patterns of routing were observed with apparent maximum densities through the center of the network.

It is evident that the combination of interference and routing concentration reduces the overall performance of the network and may contribute to congestion at nodes particularly in, or adjacent to, the center of the network topology. It is this combination of effects that leads to network throughputs in Figures 8-1 to 8-4, which are essentially equal to that which would result from a single shared medium network using a medium with a bandwidth equal to that of a single radio link.

## 9.2 Variability of source throughput

We have seen in the results from Figure 8-5 that the standard deviation of source throughput was significantly higher as pause time increased. One reason for this is simply that the throughput is higher for longer pause times thereby increasing the range of possible values that the throughput can take. More importantly, at higher pause times the network is largely static with few link changes. This can have both positive and negative effects on the sources. Stability means fewer route changes, less routing overhead and therefore higher throughput for some nodes. However, if a node is in a particularly bad position with its source being many hops from its sink, and if many nodes are attempting to route through it or through a high concentration of emitters nearby, it will receive poor performance for most if not all of the simulation. This contributed to some nodes not receiving their fair share of the resources and caused some nodes to capture and exploit this bandwidth, actually boosting overall network throughput.

However, this did not explain why so many sinks **failed to receive any data packets at all**. We examined several trace files and discovered that in nearly all trials with more than 10 sources some sources failed to deliver **any** packets to their sinks. Using the network animator and examining numerous scenarios we could not attribute the cause to any one particular topological or geometric condition such as isolation from other nodes, lack of connectivity, centrality to the topology or excessive hops. On further examination, we found that a large majority of these sinks received only the initial TCP SYN packet and no other data. This suggests that the source was able to deliver a TCP connection request but the reply never made it back to the source and no additional connection attempts were successful for the duration of the 900s simulation.

On further research into the TCP protocol, we discovered that the TCP connection protocol first employs an initial request from source to sink using a SYN packet. If no response is heard in six seconds, a second reply is sent and again 24 seconds later. The protocol then calls for a 48 second wait time before another request is sent. However, the protocol also specifies a maximum time of 75 seconds to establish a connection thus preventing any future attempts to establish this connection.

To further investigate this we attempted to stagger the start time of the TCP sources from 100ms to 15s. Although this reduced the number of non-starters, in some cases by 20%, it did not prevent all of them. We further examined the TCP settings for the receive window and discovered that the common setting of 32 Maximum Transmission Units for the max window size was most likely contributing to the problem. Since each mobile node has a 50 MTU interface queue, the node's source can take up to 32 of these slots almost immediately. Simultaneously, the routing protocol is adding routing packets and the link layer may be adding ARP requests. Subsequently, when other neighbor

nodes attempt to forward packets the queue can quickly become congested. Since routing packets are given priority in this queue the routing protocol continues to function. TCP connection requests are not so fortunate and will be dropped by the queue.

To test the theory we tried two approaches the first was to reduce the TCP congestion window to one. **This eliminated in almost all cases the appearance of non- starting nodes.** It also had only a limited effect on the overall throughput of the network. The second approach was to increase the queue size. However, neither slight increases (25 and 50 packets) nor very large increases (3000 packets) in queue size prevented non-starters and actually caused more non-starters. This is most likely caused by further delays induced by the long queues and possible domino effects on the other layer protocols.

In response to discovering these characteristics of ad hoc networks using TCP traffic, we went back to published performance evaluations to see what others reported. However, we did not find any ad hoc network evaluation that simulated FTP connections over TCP. In fact, we discovered that in almost all protocol comparisons, the authors used constant bit rate UDP sources. Since none of these papers revealed throughput or fairness issues, it raised the question of whether the characteristics discovered in our simulation with TCP traffic would also appear in simulations using CBR UDP traffic. We hypothesized that if simulations with CBR traffic sources did not produce poor network throughput or wide variations in source throughput then these characteristics found in the TCP simulations would be attributable to TCP connection and congestion control mechanisms.

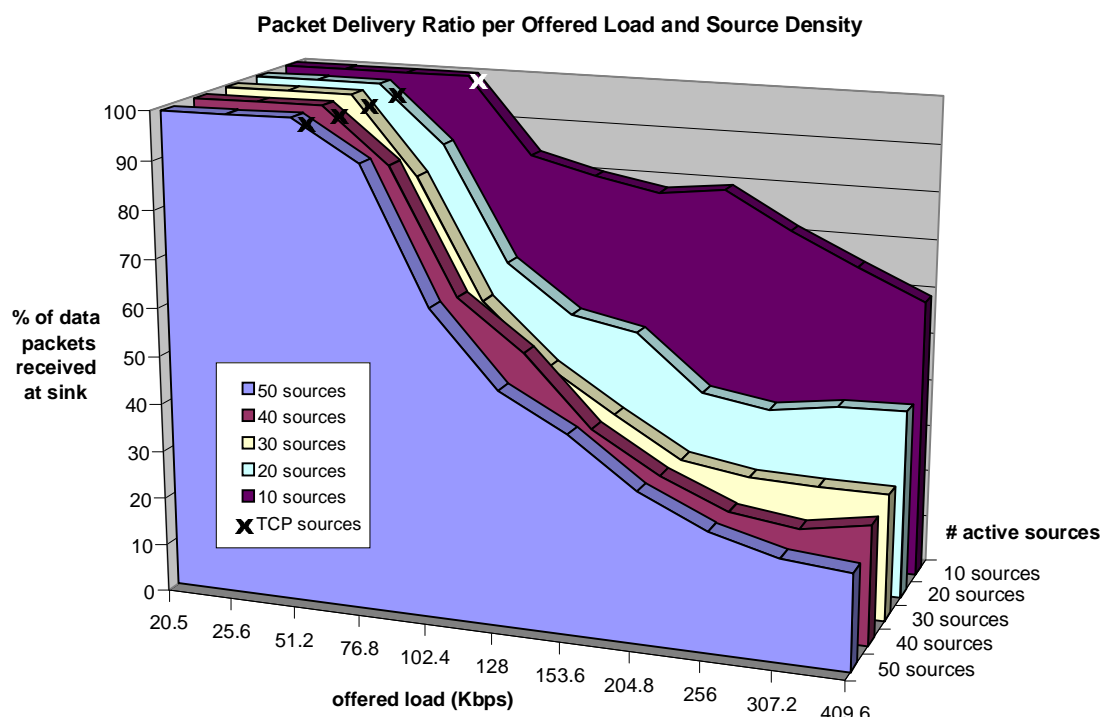
To test this hypothesis we performed individual trials with CBR sources using the AODV protocol. We ran tests of our 50-node network that increased the offered load from 20 to 400 Kbps, which was far below the shared medium maximum of 1Mbps. We also varied the density of the active sources from which this offered load was sent.

Given prior presentation of packet delivery ratios we expected to find that the throughput would be slightly less than the offered load. However, what we found instead was that when we increased the packet send-rate the network throughput was much less than the offered load. The realized network throughput for tests using AODV with our 50-node network is shown in Table 9-1. At low offered loads the throughput is as expected for all sources. However, at offered loads higher than 100 Kbps the throughput significantly decreases.

| # Sources         | Offered Load in Kbps |      |      |      |       |      |       |       |      |       |       |
|-------------------|----------------------|------|------|------|-------|------|-------|-------|------|-------|-------|
|                   | 20.5                 | 25.6 | 51.2 | 76.8 | 102.4 | 128  | 153.6 | 204.8 | 256  | 307.2 | 409.6 |
| <b>50 sources</b> | 20.4                 | 25.4 | 51.1 | 70.9 | 65.7  | 62.1 | 62    | 64.1  | 63.9 | 65.7  | 66    |
| <b>40 sources</b> | 20.4                 | 25.4 | 51.1 | 67.2 | 64.2  | 67.3 | 60.2  | 61.6  | 60   | 56.7  | 52.2  |
| <b>30 sources</b> | 20.4                 | 25.5 | 51.1 | 64.2 | 60.1  | 58.8 | 57.4  | 53.9  | 43.6 | 35.5  | 20.2  |
| <b>20 sources</b> | 20.4                 | 25.5 | 51.1 | 67.6 | 65.1  | 64   | 53.9  | 28.6  | 33.6 | 42.4  | 57.2  |
| <b>10 sources</b> | 20.4                 | 25.5 | 51.1 | 46.7 | 23.3  | 20.6 | 32.5  | 38.6  | 40   | 54.5  | 65.4  |

**Table 9-1 Throughput of CBR sources varying offered load**

Another interesting discovery was that the **packet delivery ratio** decreased significantly after an offered load of between 50 and 100 Kbps **contrary to previous published results**. In [BRO] the authors reported that the packet delivery ratio was independent of offered traffic load. Figure 9-3 shows the PDR results from our simulations. For comparison, we also marked with a bold “x” the point at each source density where TCP performance rests. Through TCP’s congestion control mechanisms, the network will maintain packet delivery ratios just at the knee of decline found with CBR sources. It does this, while maintaining throughput values more than an order of magnitude greater than that of CBR sources.



**Figure 9-3 CBR and TCP PDR per offered load and source density**

Unfortunately, as we discovered in the previous section, the benefit of improved throughput by TCP sources comes with a fairness penalty. Since fairness of CBR sources was not reported in previous papers, we also wanted to study the fairness of CBR sources. Examining Figure 9-3, we would expect that given the significant impact of congestion on CBR traffic that a network with CBR sources may also have fairness issues. To see just how unfair TCP is compared to CBR we used a commonly used measure of fairness and evaluated the trials used in the previous PDR analysis. The fairness measure taken from [STA1] is a normalized measure of dispersion of the values of  $x_i$  where in this case  $x_i$  is each source’s throughput during the trial and  $V$  is the number of sources in the trial.

$$\text{Fairness} = \frac{(\sum x_i)^2}{(V \times \sum (x_i^2))}$$

70

Figure 9-4 shows the results of this analysis. The fairness of the CBR sources comprises the surface plot and the TCP fairness values for each source density are identified with a bold “x”. It is very evident that TCP is less fair than CBR at low loads. However, as load and source density increases, CBR traffic routes experience congestion and become progressively unfair.

From Figure 9-4 the point at which CBR source fairness significantly declines is at less than 100Kbps offered load. Throughput results from these same trials showed that at this offered load network throughput was half the offered load or 50 Kbps. This is an order of magnitude less than what TCP sources delivered. It is important to note that the TCP fairness value calculations used for this graph include both the sources for which we expect TCP to not deliver its fair share and for those that never establish a connection due to the connection timeout discussed earlier. cursory tests performed have shown that eliminating that handicap would improve the overall fairness by approximately 10-20%.

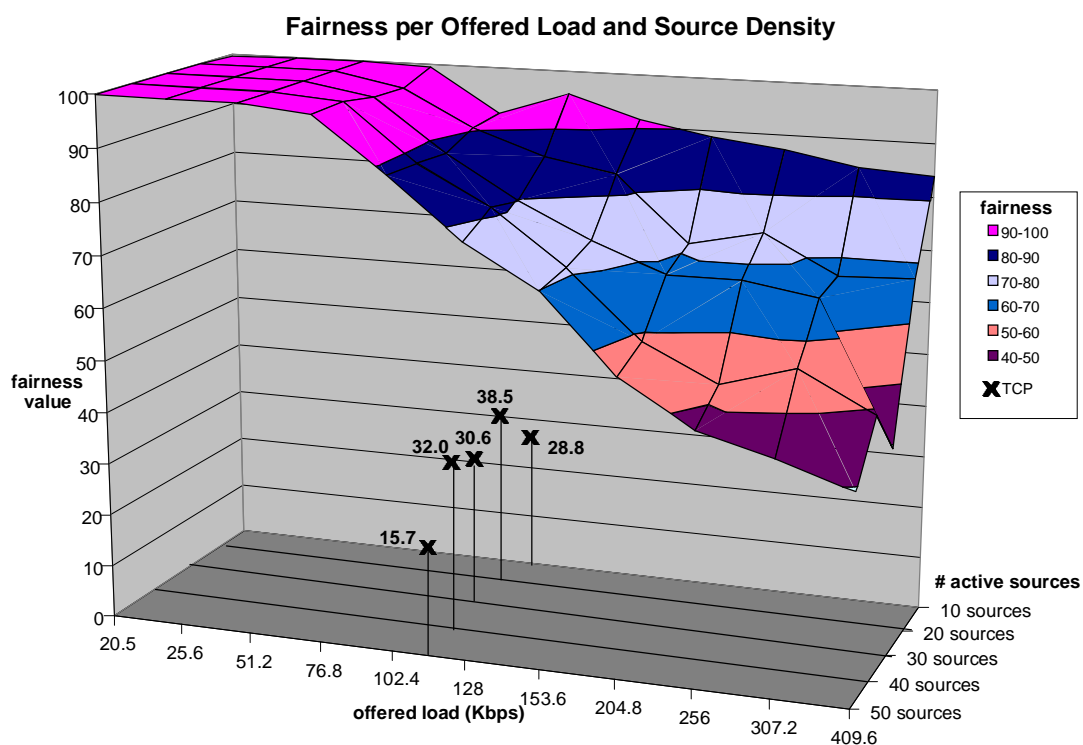


Figure 9-4 TCP and CBR fairness per offered load and source density

### 9.3 Interaction of protocol layers

A fundamental problem with ad hoc networking would appear from our analysis to be that the protocols at the various layers are predisposed to certain

actions and performance goals regardless of the effect that the pursuit of these goals have on the overall network performance as seen by the user. The transport layer protocols are primarily concerned with reliable delivery and achieving the greatest throughput while network layer protocols pursue efficient and accurate routing. They focus on shortest paths, low hop counts and sometimes consider energy levels, but are not effective at load balancing or congestion avoidance. The MAC layer focuses on reducing collisions but often at the expense of excessive delay and dropped packets. These are valid and noble pursuits in isolation but the mechanisms that accomplish these goals are often contradictory to other layer mechanisms.

## 10 Conclusions

From this research, we have seen that DSR slightly outperformed the other protocols in TCP traffic tests and has significantly less routing overhead in bytes. It also incurs a much lower packet overhead than AODV. However, DSR like the other protocols is not without limitations. Fairness remains an issue for all protocols when using TCP or when attempting to offer more than 10% of the common channel throughput with CBR sources. Fairness for DSR seems to be even a greater concern with the std. dev. of source throughput being twice that of AODV and DSR. AODV showed marginal improvement over DSDV without the excessive std. dev. of source throughput. We saw that AODV had high packet overhead but in terms of routing bytes, it was less burdensome than DSDV.

If we change focus from protocol comparison to an objective evaluation of throughput, bandwidth utilization, and quality of service issues, all existing ad hoc protocols have equally abysmal performance. These results lead us to the conclusion that the selection or development of a routing protocol is not the solution to the ad hoc network woes and further pursuit of improved routing is akin to shuffling deck chairs on the Titanic. The performance of any of the proposed routing protocols is sufficiently efficient at routing packets to not affect bandwidth. The real problems lie in the general notion of least distance routing and queuing discipline issues with respect to connection initiation. Furthermore, there are fundamental problems with transport, network and MAC layer protocols that are intensified by their myopic view of good network performance. The interaction of these protocols' mechanisms is often contradictory. To obtain good performance in MANETs will require a reexamination of the TCP protocol startup procedure, routing protocol load balancing and route selection mechanisms and MAC layer collision avoidance schemes. A collective approach that capitalizes on sharing of information, synchronization of actions, and complimentary mechanisms is needed.

VMTS is a step toward this approach by exploring the concept of MAC layer routing. However, modifications to VMTS that circumvent the same occurrences of throughput, bandwidth and QOS issues found in the other protocols are needed. The realization of these issues led us to a premature termination of VMTS implementation, but this in turn, allowed us to further investigate the causes of the poor performance. Additional mechanisms to consider include modifying the least hop-count decision to incorporate delay or congestion information in the path selection metric in an effort to improve load balancing. Additional use of MAC layer feedback and transport layer congestion windows would serve to unify the layers and improve performance. Dynamic power control based on neighbor sets to find the balance between appropriate range and minimizing interference is essential. Moreover, whenever practical, the ability to dynamically adjust routing constants based on environmental factors would provide a more responsive protocol.

A secondary but probably equally as important realization is that the TCP startup procedure should be revised for use in ad hoc networks or at the very

least, priority should be given to TCP SYN packets just as it is given to routing packets. Although generic TCP is inherently unfair, it is much worse in ad hoc networks, particularly in those with a demanding traffic load. Although some might argue that the traffic pattern used in our model is unusual and it is artificial to suggest that all sources would begin transmitting at the same time, our experiments have shown that the phenomenon in question is robust to spacing of the connection attempts at least at the time scale of practical user interactions. I would further argue that having personally experienced the sudden surge of transmissions in a military voice radio network when a significant event occurs that this is not so unusual. This analogy could easily translate to data networks that consist entirely of sensors or terminals exchanging situational awareness information. On any sudden occurrence of a large magnitude, the automated preprogrammed instruction sets and subsequent transmissions would not be curbed by good radio procedures or the patience of an experienced operator.

Therefore, any future analysis of ad hoc networks must include TCP traffic, or a combination of traffic sources. It must also include higher offered loads and mobility. Finally, reporting of performance statistics should also include analysis of the point at which the network fails. It is from knowledge of this point that one realizes that current approaches are not effective.



## REFERENCES

- [BAK] F. Baker. "An outsider's view of MANET", IETF Network WG. March 2002.
- [BRO] J. Broch, D. Maltz, D. Johnson, Y. Hu and J. Jetcheva. "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", *ACM/IEEE the 4<sup>th</sup> Annual Int. Conf. on Mob. Comp. And Net. (MOBICOM'98)*. Oct 1998.
- [CAM] Tracy Camp, Je. Boleng, Brad Williams, Lucas Wilcox, William Navidi. "Performance Comparison of Two Location Based Routing Protocols for Ad Hoc Networks". IEEE. FEB 2002.
- [CEL] E. Celebri. Masters Thesis. "Performance Evaluation of Ad Hoc Protocols". <http://www.cmpe.boun.edu.tr/~emre/research/msthesis/esim.html>. 2001.
- [COR1] S. Corson and J. Macker. "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, RFC2501, Request for Comments". January 1999.
- [COR2] S. Corson, J. Macker, and G. Cirincione. "Internet Based Mobile Ad Hoc Networking, *IEEE Internet Computing*. Jul-Aug 1999.
- [FAL] K.Fall, K. Varadhan eds. "The ns Manual". The VINT Project, UC Berkeley, LBL,USC/ISI, and XEROX PARC. available from <http://www.isi.edu/nsnam/ns/ns-documentation.html>, 10 September, 2002.
- [IEE] IEEE. IEEE 802 Standard, *Local and Metropolitan Area Networks: Overview and Architecture*. 1990
- [IET] IETF, MANET Working Group. "MANET-Charter", <http://www.ietf.org/html.charters/manet-charter.html>. Aug 2001.
- [JOH] P.Johansson, T.Larsson, N.Hedman, B.Mielczarek, and M.Degermark. "Routing Protocols for Mobile Ad-Hoc Networks - A Comparative Performance Analysis". Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom).1999.
- [PAH] K. Pahlavan and P. Krishnamurthy. "Principles of Wireless Networks, A Unified Approach".Upper Saddle River, NJ: Prentice Hall PTR, 2002. pp xi-3.
- [PER] C. Perkins. "Ad Hoc On Demand Distance Vector (AODV) Routing", Internet draft, draft-ietf-manet-aodv-00.txt, November 1997.
- [STA] W. Stallings. *Local and Metropolitan Are Networks: Sixth Edition*. Upper Saddle River, NJ: Prentice-Hall Inc., 2000.
- [STA1] W. Stallings. *High Speed Networks: TCP/IP and ATM Design Principles*. Upper Saddle River, NJ: Prentice-Hall Inc., 1998.
- [SU] W. Su, S. Lee and M. Gerla. "Mobility Prediction an Routing in Ad Hoc Wireless Networks", *International Journal of Network Management*. November 2001.
- [US1] U.S., Department of the Army. DTOE 17315L, *Tank Battalion Heavy Division*. Washington DC: Government Printing Office. April 1997.

- [US2] U.S., Department of the Army. FM 7-8, *The Infantry Platoon and Squad*. Washington DC: Government Printing Office. 31 December 1984.
- [US3] U.S., Department of the Army. FM 7-10, *The Infantry Rifle Company*. Washington DC: Government Printing Office. 14 December 1990.
- [US4] U.S., Department of the Army. FM 7-20, *The Infantry Battalion*. Washington DC: Government Printing Office. 28 December 1984.
- [ZYS] Q. Bi, G. Zysman and H. Menkes. "Wireless Mobile Communications at the Start of the 21<sup>st</sup> Century", *IEEE Communications Magazine*. January 2001.